

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Adaptation du système "Auteur !" pour la réalisation d'exercices destinés aux personnes handicapées

Platteeuw, Pascal

Award date:
1994

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

*Facultés Universitaires Notre-Dame de la Paix
Institut d'Informatique
rue Grandgagnage, 21, 5000 Namur*

**Adaptation du système "Auteur !" pour la réalisation
d'exercices destinés aux personnes handicapées.**

Pascal PLATTEEUW

Promoteur : Madame le Professeur Monique NOIRHOMME - FRAITURE

Mémoire présenté en vue de l'obtention du grade de
Licencié et Maître en Informatique

Année académique 1993 - 1994

Remerciements

Je tiens à remercier toutes les personnes qui m'ont aidé pendant cette dernière année :

Tous ceux qui m'ont accueilli dans les différents lieux de stages : M. LAMY et son équipe, M. DONEGAN et son équipe, pour leur accueil chaleureux et leur sympathie;

Luc VANDENABEELE pour sa patience et ses nombreuses explications;

A tous ceux qui ont supporté mes humeurs (parfois massacrantes) pendant toute une année : mes parents, mon frère, Nathalie,...

Enfin, Madame le Professeur Monique NOIRHOMME-FRAITURE, sans qui ce mémoire n'aurait pas eu lieu, pour ses nombreux conseils et sa bonne humeur permanente.

Résumé

La première partie de ce travail a consisté en une évaluation des capacités et des outils du logiciel "Auteur !" [VAND,93]. Les éducateurs et les thérapeutes se sont rendus compte de ce que ce programme pouvait leur apporter. Ils ont aussi constaté certaines carences, notamment au niveau des moyens d'interaction et des outils. Enfin, ils nous ont exprimé leurs désirs quant au programme en lui-même. Ces remarques ont conduit à une demande d'ajout d'outils au logiciel, et ce afin de pouvoir réaliser des exercices de latéralité, ce qui était leur but premier.

Ensuite, dans une seconde partie, nous avons, dans la mesure du possible, ajouté les outils demandé au logiciel "Auteur !". Une partie existait déjà, mais il a fallu la compléter notamment des points de vue : analyse de la trace, sélection d'objets à l'écran, aide à l'utilisation, contrôle du programme ,...

Dans un souci de continuité, les outils ajoutés ont été réalisés selon une méthode "orientée objet". L'implémentation de ces outils s'est faite en **C++** sous l'environnement **Microsoft Windows**.

Abstract

The first part of this work was the evaluation of the capacity and the tools of "Auteur !" [VAND,93]. Educators and therapists look at what this program can bring to them. They also establish some deficiency, more particularly at the level of the means of interaction and at the level of the tools. Finally they told us some wishes for the program itself. Those ones lead to an adding of tools in the program and this, to make possible the realisation of laterality exercises which was their first aim.

In a second part, we've added, as far as possible, the asked tools in the program "Auteur !". One part of those tools was already existing. But we've to complete it more particularly for the means of interaction, the analysis of the trace,...

In order to manage with what was done, those tools were implemented according to an object oriented method. It was done in **C++** under **Microsoft Windows** environment.

Table des matières

Résumé - Abstract

Table des matières

Introduction

1. Un bref rappel **1**

1.1. Introduction	2
1.2. Les outils déjà implémentés	2
1.2.1 L'écran	2
1.2.2 L'icône	2
1.2.3 L'animation	2
1.3. Un exemple d'exercice	5

2. La population visée **8**

2.1. Introduction	9
2.2. Les problèmes à traiter	9
2.3. La latéralité	9
2.3.1 les troubles de la latéralité	13
2.3.2 L'ordinateur comme aide	15
2.4. La structuration spatiale	15
2.4.1 Les troubles de la structuration spatiale	17
2.4.1.a Premier symptôme : l'enfant ignore les termes spatiaux	17
2.4.1.b Deuxième symptôme : l'enfant connaît les termes spatiaux mais perçoit mal les positions	18
2.4.1.c Troisième symptôme : l'enfant perçoit bien l'espace qui l'entoure mais s'oriente difficilement	18
2.4.1.d Quatrième symptôme : l'enfant s'oriente bien mais n'a pas de mémoire spatiale	19
2.4.1.e Cinquième symptôme : l'enfant n'a pas d'organisation spatiale	19
2.4.1.f Sixième symptôme : l'enfant n'acquiert pas la réversibilité et la transposition	19

2.4.1.g Septième symptôme : l'enfant a des difficultés à comprendre les relations spatiales	20
2.4.2 Moyen de rééducation	20
2.4.2.a Du plus simple au plus complexe	20
2.4.2.b L'ordinateur et la structuration spatiale	22
2.5. L'orientation temporelle	22
2.5.1 La mise en place de l'ordre et de la succession	24
2.5.2 La découverte de la durée et des intervalles	24
2.5.3 L'apprentissage des cycles et du renouvellement cyclique des périodes	25
2.5.4 L'apparition du rythme	25
2.5.5 Les troubles de l'orientation temporelle	26
2.5.5.a Premier symptôme : l'enfant est incapable de retrouver l'ordre et la succession des événements	26
2.5.5.b Deuxième symptôme : l'enfant ne perçoit pas les intervalles	26
2.5.5.c Troisième symptôme : l'enfant n'a pas un rythme régulier	26
2.5.5.d Quatrième symptôme : l'enfant n'a pas la notion d'heure et ne peut organiser son temps	26
2.5.6 Moyens de rééducation	26
2.5.7 L'ordinateur et l'orientation temporelle	27
2.6. Les Infirmités Motrices Cérébrales (IMC)	27
2.7. Les enfants autistes	28
2.8. Conclusion de ce chapitre	29
3. Expériences extérieures	30

3.1. Introduction	31
3.2. Quelques jours en Angleterre	31
3.2.1 Qu'utilisent-ils comme matériel ?	31
3.2.2 Quelle est (quelles sont) leur(s) méthode(s) de développement ?	32
3.2.3 Quels sont leurs projets ?	33
3.2.4 Paramétrage de logiciels	33
3.2.4.a AccessMath	33
3.2.4.b StartWrite	34
3.2.4.c Switch Access for Windows (SAW)	35
3.3. Le stage à Bruxelles	37
3.4. Evaluation de logiciels éducatifs	38
3.4.1 Huit programmes testés dans différentes classes	38
3.4.2 En complément d'information	43
3.5 En conclusion	46

4. Quoi de neuf ?

47

4.1. Introduction	48
4.2. Les besoins rencontrés	48
4.2.1 Les besoins impératifs	48
4.2.2 Les besoins secondaires	49
4.2.3 Les besoins facultatifs	49
4.3. Les solutions proposées	50
4.3.1 Les solutions proposées pour les besoins impératifs	50
4.3.1.a Le sauvetage et le rechargement	50
4.3.1.b Le système de coordonnées à l'écran	51
4.3.1.c Le détournement d'objet	53
4.3.1.d Le réglage des moyens d'interaction	55
4.3.1.e Le commentaire du code de monsieur Luc Vandenaabeele	58
4.3.2 Les solutions proposées pour les besoins secondaires	58
4.3.2.a L'analyse de la trace	58
4.3.2.b Le mode d'emploi	58
4.3.2.c La compatibilité d'écrans	59
4.3.2.d La navigation d'objet	59
4.4. Conclusion de ce chapitre	60

5. Outils et difficultés de programmation

61

5.1. Windows	62
5.2. La programmation Windows	62
5.3. Les difficultés rencontrées	65
5.3.1 La sauvegarde et le chargement	65
5.3.2 Le déplacement d'objet	67
5.3.3 Les moyens d'interaction	67
5.3.4 La ligne et le texte	68
5.3.5 Le détournement	68
5.3.6 Les circuits	68
3.7 Le joystick	69
4. Conclusion de ce chapitre	70

6. Une méthode de construction d'exercices **71**

6.1. Introduction	72
6.2. A vos crayons	72
6.2.1 Un effort d'imagination	72
6.2.2 Un plan sur papier	72
6.3. Passons sur la machine (un exercice mono-écran)	73
6.3.1 Une brève terminologie	73
6.3.2 Présentation des outils	76
6.3.3 Conception de l'exercice sur la machine	78
6.4. A vos crayons (un exercice multi-écrans)...	81
6.5. Passons sur la machine (un exercice multi-écrans)...	82
6.6. Conclusion de ce chapitre	86

7. Structures de données **87**

7.1. Introduction	88
7.2. La programmation orientée objet : un outil bien adapté	88
7.3. Structures de données des éléments préexistants	91
7.3.1 L'objet "élément"	91
7.3.2 L'objet "écran"	92
7.3.3 L'objet "icône"	92
7.3.4 L'objet "animation"	93
7.3.5 L'objet "TAction"	93
7.3.6 L'objet "TListeActions"	94
7.4. Structures de données des éléments ajoutés	94
7.4.1 Le texte	94
7.4.2 La ligne	95
7.4.3 Le circuit	95
7.5. Conclusion de ce chapitre	96

8. Perspectives	97
<hr/>	
1. Introduction	98
2. Des ajoutes	98
2.1 L'aide en ligne	98
2.2 Les séquences animées	98
2.3 CD-ROM et commande vocale	99
2.4 L'analyse de la trace	99
3. Des améliorations	100
4. En conclusion de ce chapitre	100
 Conclusion	 101
<hr/>	
Bibliographie	103
<hr/>	
Annexe A : Le mode d'emploi du programme	105
<hr/>	

Introduction

Les Facultés au service du handicap

Depuis quelques années, les Facultés Universitaires Notre-Dame de la Paix à Namur reçoit des demandes émanant de différents centres pour personnes handicapées.

Sous la direction de Madame Noirhomme, plusieurs de ces demandes ont déjà été réalisées. Ainsi, les logiciels **Ordi-Théâtre**¹, **Comptes**², **Corps Humain**³ permettent à des personnes défavorisées de faire des choses qu'ils ne pourraient pas faire autrement. Ils peuvent jouer aux marionnettes, tenir un budget, étudier le corps humain avec le plus patient des professeurs : l'ordinateur.

*Le travail de Stéphanie et Rudy*⁴

Le travail de Stéphanie et Rudy, point de départ du travail de l'année passée et de cette année, consistait à établir une boîte à outils avec laquelle la conception d'exercices destinés aux enfants handicapés serait facilitée. Cela de manière à pouvoir accélérer le travail d'un informaticien pour concevoir les exercices sur ordinateur.

Pour y parvenir, ils ont d'abord repéré quels concepts se retrouvaient dans toutes (ou presque) les applications destinées aux personnes handicapées. Une fois ces concepts repérés, ils ont été assemblés dans une boîte à outils. A ce stade chaque outil était implémenté séparément en langage C.

*Le travail de Luc*⁵

Le travail de Luc, quand à lui, consistait à reprendre cette boîte à outils, à l'évaluer sur le terrain puis à intégrer tous les outils ainsi validés au sein d'un même programme. Le choix de l'environnement **Windows** a été fait pour des raisons d'ergonomie d'écrans et de facilité de multimédia (ce dernier concept ayant été ajouté lors de l'évaluation.).

Le logiciel "Auteur !" ainsi créé reprenait tous les outils mis en évidence par Stéphanie et Rudy et rencontrait le but fixé, à savoir, se passer d'un informaticien pour pouvoir concevoir des exercices sur un ordinateur.

Le choix de l'environnement **Windows** ainsi que certains problèmes rencontrés ont imposé la réécriture de la boîte à outils en langage C++.

¹ Cfr [BROU,90].

² Cfr [DEPL,89].

³ Cfr [LEP,90].

⁴ Cfr [DEMO,92]

⁵ Cfr [VAND,93]

Le projet de cette année

Le projet de cette année consiste, dans un premier temps, à élaborer certains exercices en collaboration avec un centre de Bruxelles, le centre "La Famille". Ces exercices doivent servir de base au travail de cette année. Il faut voir si, dans l'état actuel, le programme est satisfaisant du point de vue des outils proposés ou s'il faut en ajouter. Les éventuelles ajoutes et rectifications feront l'objet de ce mémoire.

Dans un premier chapitre, nous proposerons un bref rappel des différents outils déjà implémentés dans le logiciel "Auteur !" développé par Luc Vandenabeele. Nous donnerons également un très bref aperçu des exercices et actions possibles avec ce programme.

Dans un second chapitre, nous définirons ce que sont exactement les problèmes qu'il nous faudra traiter cette année. Ces problèmes sont : les troubles de la latéralité, les troubles de la perception et de la situation spatio-temporelle. Dans ce même chapitre, nous dirons à qui s'adresse ce genre de logiciel et les utilisations que l'on peut en faire.

Le troisième chapitre présentera, outre les expériences des lieux de stage (en Angleterre et en Belgique), différents programmes destinés à l'éducation. Notons immédiatement que tout ces programmes ne sont pas spécialement adaptés au domaine du handicap. Toutefois, certains peuvent servir, en modulant un peu la difficulté.

Comme plusieurs éléments sont destinés à entrer dans le logiciel, il nous a paru important de consacrer un chapitre à ces nouveautés. Ce sont surtout les besoins exprimés par les utilisateurs et les solutions apportées qui seront développées dans ce quatrième chapitre.

La programmation de tout ce petit monde, ne peut se faire sans difficultés. Aussi pour que chacun voie bien à quelles difficultés nous avons été confrontés, nous présenterons les outils avec lesquels nous avons travaillé. Nous verrons également les bons cotés de ces outils.

Ensuite, nous présenterons une ébauche de méthode de conception d'exercice avec le logiciel "Auteur !". Le but de ce sixième chapitre n'étant pas d'être complet quand aux possibilités de "Auteur !", l'exemple que nous prendrons sera relativement simple.

Nous présenterons les structures de données de ce qui avait été fait et de ce qui a été ajouté cette année dans un septième chapitre.

Les perspectives d'avenir feront l'objet d'un huitième chapitre.

En annexe, nous donnerons un mode d'emploi détaillé du logiciel. Celui-ci, ainsi que la méthode reprise au sixième chapitre seront les deux documents de base pour l'utilisateur d' "Auteur !".

Chapitre 1

Un bref rappel

Où l'on parle de ce qui a déjà été fait dans le cadre du projet "Auteur !" et où l'on découvre ce qu'il est possible de faire avec ce logiciel et les outils déjà implémentés.

1.1. Introduction

Le logiciel "**Auteur!**", tel que nous l'avons reçu en septembre 1993, n'était pas terminé. Toutefois, certains outils étaient déjà présents et tout à fait opérationnels. D'autres outils attendaient le bon vouloir d'un programmeur. C'est à ce moment que nous avons repris le programme...

1.2. Les outils déjà implémentés

Les outils déjà présents étaient les outils fondamentaux sans lesquels nous n'aurions rien pu faire. Citons-les dans l'ordre d'importance :

- **l'écran** : c'est sur cet outil que viennent se greffer les autres. Il va de soi que sans cet outil, l'éducateur ne pourrait même pas commencer son travail. Il était donc un des éléments les plus importants. Et comme de juste, il a été, nous croyons, l'un des premiers outils implémentés par Luc Vandenaabeele.

Un écran est défini par un nom, une couleur et un nombre d'éléments contenus. Le nom et la couleur se changent très facilement par un double-click sur l'écran lui-même. Par cette action, l'utilisateur fait apparaître une boîte de dialogue contenant un champ d'édition et quatre boutons. Deux de ces boutons servent à accepter ou à refuser les changements. Un troisième bouton permet de changer la couleur de fond de tout l'écran, le quatrième permet de définir des actions que l'on associe à cet écran. Le champ d'édition sert à changer le nom de l'écran afin de lui donner un nom significatif. En effet, ce nom est par défaut "ECRAN x" où x est un numéro d'ordre. Il vaut mieux pour l'utilisateur qu'il donne à cet écran un nom qui a du sens. La couleur par défaut est le blanc.

- **l'icône** : concept clé du logiciel, l'icône est certainement un des outils les plus utiles si pas le plus utile. A partir de l'icône, on peut quasi tout définir. D'après les observations que nous avons pu faire, c'est de loin l'outil le plus utilisé. Peut-être est-ce dû aussi au fait qu'il répond parfaitement aux différentes actions qu'on voudrait lui faire entreprendre.

L'icône est spécifiée par un nom, une taille, un motif (dessin) et certains attributs. Ces derniers précisent si l'icône est sélectionnable, si elle est visible, si elle est amovible et enfin, si elle est empilable. Quand on choisit l'outil "icône" et que l'on clique sur l'écran, le motif par défaut apparaît. L'icône est dessinée, avec une taille donnée, à l'endroit où l'utilisateur a cliqué. Tous les changements de caractéristiques se font dans une boîte de dialogue que l'on obtient par un double click sur l'icône. Cette boîte de dialogue permet de changer le nom, la taille, les attributs, le pictogramme,...

- **l'animation** : l'animation sert surtout à réaliser des renforcements positifs et/ou négatifs. C'est donc aussi une clé de voûte dans un logiciel destiné à des personnes handicapées. On l'utilise toutefois très peu comme un objet en accès direct. C'est-à-dire que l'on ne met généralement pas une animation comme objet sélectionnable, ou, si c'est le cas, aucune action n'est prévue dans le cas d'une sélection de cet objet.

L'animation est caractérisée par un nom, une taille et certains attributs (tout comme l'icône) mais plus important que tout, l'animation n'est pas constituée d'un seul pictogramme. Pour définir une animation, il faut remplir un tableau (maximum 10 éléments) avec des images, des dessins. Ces images seront enchaînées plus ou

moins vite et constitueront une animation à la manière d'un film d'animation constituée de X photos enchaînées. Les paramètres de l'animation se changent de la même manière que ceux de l'icône : un double click sur le pictogramme par défaut de l'animation placée.

Associées à ces outils, il y a différentes actions. Toutes ces actions ne fonctionnaient pas au départ. Quand nous avons repris le programme, seules les actions de changement d'écran, d'exécution d'une animation et d'écoute d'un son étaient opérationnelles. Il manquait donc deux actions : attendre et positionner un objet. Celles-ci étaient prévues dans le logiciel, mais, par manque de temps, n'avaient pas été implémentées. Revoyons ensemble toutes ces actions :

- **Changer d'écran** : Cette action permet, en cas d'exercice multi-écrans (c'est-à-dire, composé de plus d'un écran), de passer d'un écran à l'autre. Par exemple, lorsqu'on clique sur la bonne icône dans l'exercice de l'"intrus", on peut passer à un autre écran qui indique un renforcement positif, et de cet écran, revenir au premier écran, pour une répétition de l'exercice ou aller vers un troisième écran pour passer à un autre exercice. Ces changements d'écran sont paramétrables par l'intermédiaire d'une boîte de dialogue. Les changements d'écran sont simplement caractérisés par le nom de l'écran à appeler.
- **Jouer un son** : Jouer un son peut être une action très importante. En effet, quoi de plus important que d'avoir un feed-back auditif. Il faut en effet prévoir que certaines personnes n'ont pas la possibilité de comprendre un exercice grâce à un texte qui explique ce qu'il faut faire. D'où la possibilité d'insérer du son dans les exercices. Les sons peuvent être également très importants dans les renforcements. N'importe quel son enregistré au format ".WAV" est jouable par "Auteur!". Le nombre de fois que l'on joue le son est lui aussi réglable.
- **Exécuter une animation** : C'est la dernière action prévue par Luc Vandenabeele. Elle consiste à enchaîner une suite d'images à la façon d'un dessin animé. Cette suite d'images est généralement constituée des images symbolisant un mouvement existant. Par exemple, une main qui s'ouvre peut être décomposée en une série de 4 ou 5 images. Le nombre de fois qu'une animation peut être jouée est entièrement paramétrable.
- **Attendre** : Attendre est une nouvelle action. Elle était prévue dans le logiciel de base, mais non implémentée. Cette action stoppe l'exécution d'un exercice pendant un certain temps ou en attendant que l'utilisateur fasse quelque chose. (Par exemple, on peut attendre que l'utilisateur presse une touche du clavier ou un bouton de la souris.) Il y a trois facteurs d'attente : on attend que l'utilisateur pousse sur une touche quelconque ou on attend que l'utilisateur appuie sur un des boutons de la souris, ou enfin, on demande à l'ordinateur de faire attendre l'utilisateur pendant un temps donné. Pendant l'attente, aucune action n'est possible et pour s'en assurer, le curseur disparaît.
- **Positionner un objet** : Ici aussi, c'est une nouvelle action. Elle permet de changer la position d'un objet à l'écran. On peut ainsi définir des exercices qui doivent remettre des images à une certaine place,... L'indication de la nouvelle position de l'objet se fait par un pointage sur un écran vide. L'endroit où l'utilisateur clique est considéré comme le point où l'objet (généralement une icône) apparaîtra.

L'utilisateur n'a donc pas à se casser la tête en calculant les nouvelles coordonnées. Il n'a qu'à cliquer à l'endroit où il veut voir apparaître l'objet.

Il va de soi que la description des objets ci-dessus et de leurs actions associées est très sommaire. L'utilisateur qui veut en savoir plus peut consulter le mode d'emploi du logiciel décrit en annexe.

Chaque élément peut évidemment avoir une ou plusieurs actions associées. S'il y en a plusieurs, elles sont exécutées dans l'ordre où elles apparaissent dans la boîte de dialogue de gestion des actions. Cette boîte de dialogue est présentée ci-dessous, avec une suite d'action à entreprendre lorsqu'on clique sur une icône. Une liste de 10 (dix) actions peut ainsi être définie. Notez que cette boîte est la même, que l'on veuille définir les actions d'un écran ou d'une animation.

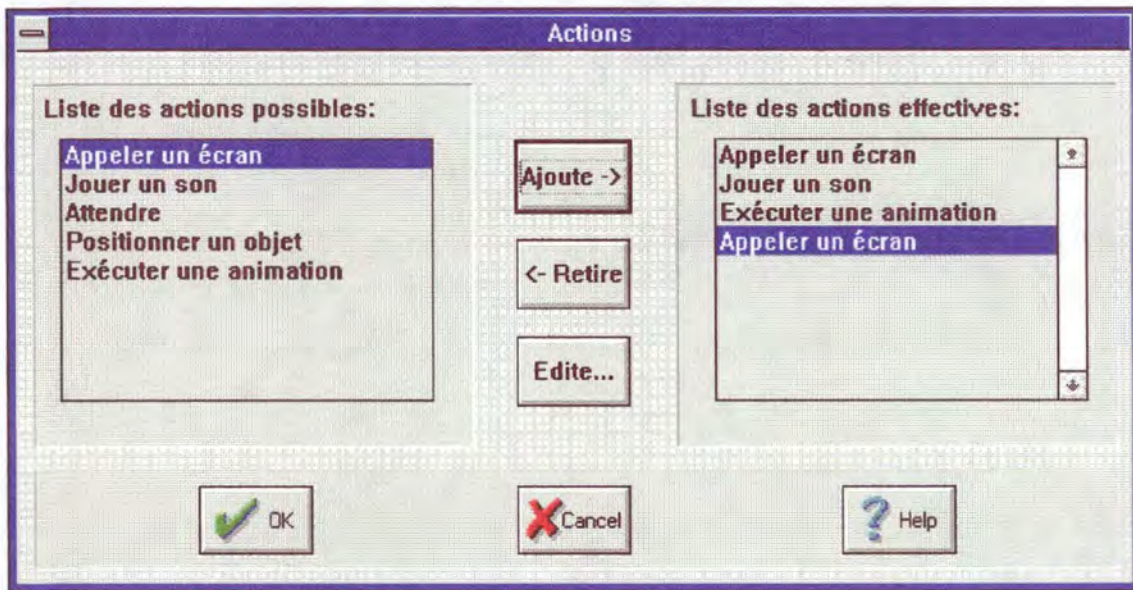


Figure 1.1. : La boîte de dialogue pour le réglage des actions d'un élément.

Les objets manipulés par l'éducateur peuvent aussi être totalement paramétrés. Ainsi, un écran peut être défini par sa couleur, son nom et le nombre d'éléments qu'il contient. Une icône est définie par un nom, une taille, une position, des propriétés et enfin, avec un fichier image (fichier BMP ou BitMap). Enfin, une animation est définie par un nom, une position, une taille et une série d'images (toujours au format BMP).

Les actions sont également paramétrables : Quand on change d'écran, il faut indiquer vers quel écran on veut aller ou alors on choisit l'écran par défaut, qui est toujours le précédent. Quand on joue un son, il faut définir le son (il y a bien sûr un son par défaut), et il faut dire combien de fois il sera joué. Si on veut faire une animation, il faut dire quelle animation il faut exécuter (tout au moins, s'il y en a plusieurs) et combien de fois il faut l'exécuter. Pour une action d'attente, il faut indiquer l'événement que l'on attend : soit un temps donné, soit un click souris, soit une frappe clavier. L'action de positionnement, elle, se définit par l'objet dont la position doit changer et par les nouvelles coordonnées qu'il doit prendre.

Tous ces paramétrages sont très faciles à réaliser. Quelques clicks bien placés de votre animal préféré et les exercices se créent quasi tout seuls.

1.3. Un exemple d'exercice

Pour terminer ce chapitre, nous aimerions présenter un petit exercice implémenté à l'aide des outils existant en septembre 1993. Pour des raisons purement techniques, les écrans présentés ci-dessous ne sont pas des copies d'écrans provenant de "Auteur !". Ce sont des reproductions d'écrans faites à la main. C'est pour cela qu'ils n'ont pas le cachet donné par "Auteur !" aux écrans conçus à l'intérieur du logiciel. Nous présenterons dans une partie ultérieure un exemple d'exercice plus complet, basé sur des copies d'écrans réelles.

Cet exercice est un "Chercher l'intrus". Nous présentons quatre icônes à un enfant. Celui-ci doit choisir l'intrus parmi ces quatre dessins. S'il réussit, un deuxième écran apparaît pour le féliciter (renforcement positif). S'il échoue, un troisième écran le lui signale (renforcement négatif).

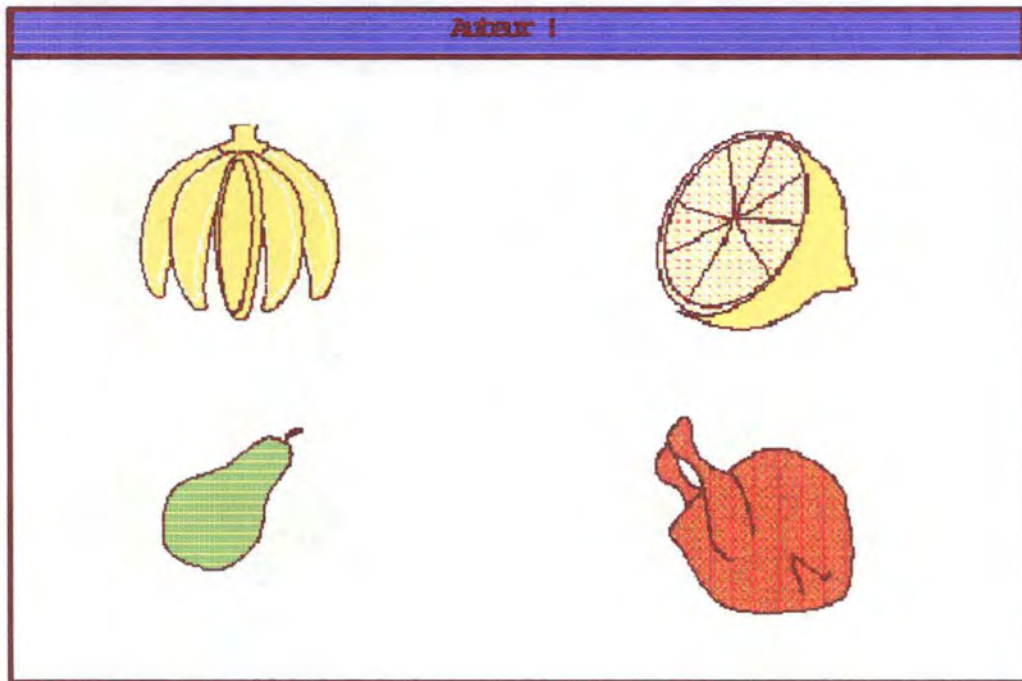


Figure 1.2. : Un exemple d'exercice réalisable par "Auteur !", Cherchez l'intrus...

Si l'enfant clique sur un des fruits, il voit apparaître l'écran suivant :



Figure 1.3. : Un écran de renforcement négatif.

Le passage de cet écran vers le premier écran peut être automatique (par exemple, après un certain temps) ou provoqué par l'enfant (par un click souris). Si l'enfant clique sur le poulet rôti du premier écran, c'est l'écran suivant qui apparaît :

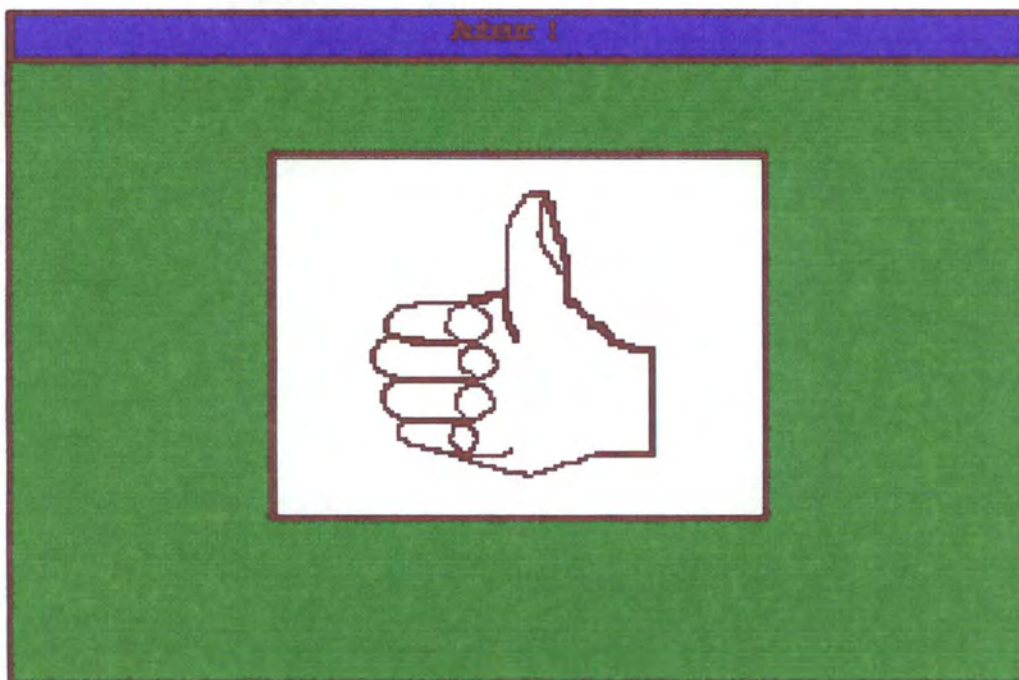
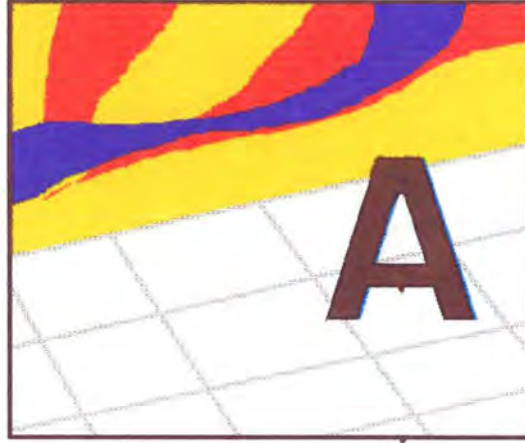


Figure 1.4. : Un écran de renforcement positif.

Notez que, sur chaque écran, on peut ajouter une icône ne faisant pas partie de l'exercice et qui est juste destinée à donner les instructions de manière auditive à l'enfant. Cette icône pourrait simplement être le dessin d'une oreille ou d'un micro, ou encore de toute autre chose qui fasse penser à l'enfant que c'est quelque chose qu'il va entendre.

En conclusion de ce chapitre, nous dirions que le programme remis par Luc Vandena-beele était déjà très complet. Nous n'avons pas eu grand mal, avec son aide, à "entrer" dans ce programme et à le compléter. Deux mots sur l'accueil reçu par ce programme : les éducateurs à qui nous l'avons soumis se sont très vite montrés enthousiastes. Ils nous ont signifié ce qui leur semblait des lacunes et nous ont transmis leurs demandes par rapport à ce projet. Cela fera l'objet d'un chapitre ultérieur.



Chapitre 2

La population visée

Où l'on décrit les problèmes des utilisateurs concernés par ce travail. Le lecteur trouvera dans ce chapitre des notions de psychomotricité dans les domaines de la latéralité, de l'orientation spatiale et de la structuration temporelle.

2.1. Introduction

Notre métier d'informaticien, et le développement d'applications "normales", ne nous sensibilisent généralement pas aux problèmes à propos desquels nous avons été sollicités cette année. Nous ne réfléchissons sans doute que très peu, quand nous créons une application, que l'utilisateur pourrait avoir un comportement différent du nôtre. Et ce pour la simple raison que nous présumons du comportement d'un utilisateur placé dans une situation particulière.

La situation est beaucoup plus délicate quand le comportement de l'utilisateur est une personne ou un enfant handicapé. On ne peut prévoir quelle réaction aura l'utilisateur face à tel ou tel stimulus. Ces gens ne réagissent pas comme des personnes dites "normales" (encore que la normalité soit un concept difficile à définir). Soit parce qu'elles n'ont pas les capacités mentales de le faire, soit parce qu'elles n'ont pas la faculté physique de le faire, soit enfin, parce qu'elles n'ont ni l'une ni l'autre.

Nous ne pouvons exclure ces gens de l'accès à la technologie informatique sous ce prétexte qu'elles ne peuvent utiliser ce que nous leur proposons. C'est à nous, informaticiens, de nous adapter à eux et non l'inverse. Pour pouvoir faire ce pas, il faut savoir de quoi ces utilisateurs ont besoin, il faut savoir quels sont leurs problèmes.

Dans le cas qui nous a occupé cette année, les personnes handicapées étaient des IMC, c'est-à-dire des Infirmités Motrices Cérébrales, tant adultes qu'enfants. La plupart de ceux-ci ont un problème en commun : il ne maîtrisent pas l'espace qui les entoure et n'ont aucun contrôle sur le temps qui sépare les actions. Leur motricité n'est donc pas bonne.

C'est de ces problèmes de motricité, et plus spécialement de psychomotricité, dont nous allons parler tout au long de ce chapitre. Nous réserverons, en fin de chapitre, une partie pour la description des enfants IMC, des enfants autistes et de leur interaction avec l'ordinateur. Nous nous sommes basés sur des ouvrages de psychologie de l'enfant et de psychomotricité pour rédiger cet article. Le lecteur intéressé trouvera plus dans [Meur,88], et dans [Rond,85], ainsi que dans

2.2. Les problèmes à traiter

Ce mémoire vise à utiliser "Auteur!" pour créer des exercices prenant en compte trois types de problèmes : les troubles de la latéralité, les problèmes d'orientation spatiale, et enfin, mais dans une moindre mesure, les problèmes de situation temporelle.

Passons en revue ces trois types de problèmes :

2.3. La latéralité

Il existe plusieurs types de latéralité. On en distingue au moins deux : la latéralité cérébrale et la latéralité corporelle.

La première s'occupe de tout ce qui est spécialisation des hémisphères du cerveau. Ainsi, on sait que, dans la plupart des cas, l'hémisphère gauche s'occupe principalement du traitement de l'information présentée en séquence linéaire (le langage en est un exemple) et du contrôle de l'hémicorps droit tandis que l'hémisphère droit s'occupe des informations multidimensionnelles (c'est-à-dire, tout ce qui est spatial) et du contrôle de l'hémicorps gauche.

La seconde notion de latéralité, la latéralité corporelle, est liée à la notion de schéma corporel. Ce schéma corporel, chez un enfant normal, se précise entre 3 et 20 mois. Vers 3 mois, l'enfant découvre ses mains et les différents mouvements qu'il peut faire avec leur aide. Viennent ensuite, dans l'ordre, la découverte du visage, des pieds,...

Vers 9 mois, l'enfant différencie ses mains et les objets qu'il touche. Ce n'est que vers 20 mois que l'enfant prend conscience de son corps. A cet âge, il sait enfin que son corps existe, que tous ses membres s'y rattachent et que ce corps n'est pas quelque chose de temporaire mais de permanent.

La latéralité corporelle est liée à ce concept en ce sens que l'enfant utilise plus un membre que l'autre. On peut distinguer plusieurs types de latéralité : celle de l'oeil, du bras, du pied... Lorsqu'on observe l'enfant, on peut détecter la prédominance d'un côté ou de l'autre.

Cette latéralité est plus ou moins marquée et elle est variable. L'enfant peut utiliser sa main gauche pour dessiner (latéralité gauche) et son pied droit pour taper dans un ballon (latéralité droite), ou utiliser indifféremment ses deux mains et ses deux pieds. On appelle alors cela de l'ambidextrie. On peut aussi parler de latéralité mixte, lorsque l'enfant utilise à la fois la main et le pied droite et son oeil gauche ou inversement.

Certains spécialistes n'hésitent pas à définir quatre (4) types de latéralisation :

- **La dextralité vraie** : c'est le cas d'une latéralisation où la dominance cérébrale est à gauche. C'est-à-dire que c'est l'hémisphère gauche du cerveau qui est prépondérant par rapport à l'hémisphère droit. Toute la motricité se fait donc à droite. C'est le cas le plus fréquent, comme le montre le tableau repris en page 11.
- **La sénestralité vraie** : c'est l'inverse, la dominance se fait ici à droite. Il est cependant faux de considérer un gaucher pur comme étant systématiquement organisé comme un droitier à l'envers.
- **La sénestralité fausse ou gaucherie** : celle-ci n'est généralement qu'accidentelle. Elle se produit quand un droitier naturel ne peut, pour des raisons matérielles, se servir de sa main droite (amputation, paralysie,...).
- **La dextralité fausse** : c'est l'inverse de la précédente. Pour une raison ou une autre, le gaucher naturel est empêché de se servir de sa main gauche. C'est alors l'hémisphère gauche du cerveau qui doit reprendre le contrôle, alors que ce n'est pas sa tendance première.

Entre 1 et 5 ans, la latéralité change chez l'enfant. Avec l'âge, la proportion de droitiers augmente par rapport au nombre d'enfants ambidextres. Le nombre de gauchers reste, proportionnellement parlant, stable.

La latéralité est, au vu des données récentes, en partie héréditaire et en partie acquise grâce au milieu et à l'expérience. Le fait qu'il existe des "gauchers contrariés" est, de ce fait, assez significatif. En effet, "les gauchers contrariés" sont souvent des gens qui ont été rééduqués de manière à "mieux fonctionner" dans une société prévue pour les droitiers.

Beaucoup d'idées fausses existent sur la latéralité :

- La première est qu'une rééducation telle que celle que nous venons d'évoquer peut provoquer un bégaiement chez l'enfant. Il nous faut dire, en rectifiant cette idée, qu'aucune preuve n'a jamais été faite, ni dans un sens, ni dans l'autre.
- La seconde est qu'il y a un rapport entre la latéralité et l'apprentissage de la lecture et de l'écriture. Encore une fois, s'il est vrai que la latéralité intervient dans ces deux derniers apprentissages, elle n'est pas la seule à intervenir. Les problèmes de dyslexie, de dysgraphie, de dysorthographe, sont beaucoup plus complexes : ils font souvent intervenir en plus, des problèmes de relations interpersonnelles et/ou des problèmes de la construction de la personnalité.
- Enfin, troisième idée fausse, la latéralité est souvent confondue avec la "connaissance gauche - droite". Celle-ci découle de la dominance latérale ([Meur,88]). L'enfant généralise la perception qu'il a de son axe corporel à tout ce qui l'entoure. La "connaissance gauche - droite" sera d'autant facilitée que la latéralité de l'enfant sera plus affirmée et plus homogène.

Le tableau ci-dessous, tiré de [Rond,85] illustre ce que nous venons de dire

niveau examiné		7 - 8	9 - 10	11 et plus
Main	D	54	61	76
	A	31	25	11
	G	15	13	13
Oeil	D	52	52	55
	A	8	9	6
	G	39	38	39
Pied	D	74	78	71
	A	7	4	5
	G	18	18	23

Figure 2.1: Pourcentage d'enfants droitiers, gauchers et ambidextre par tranches d'âge

La latéralité est liée très étroitement au développement de l'enfant. Ce n'est pas un concept qui apparaît brutalement. Elle ne s'installera vraiment bien (c'est-à-dire efficacement) que dans le cas où l'enfant aura traversé les différentes étapes particulières de son développement. La plus importante de ces étapes est sans doute la coordination oculo-manuelle. C'est en effet cette fonction qui permet à l'enfant de maîtriser sa motricité fine et sa perception visuelle. Une fois qu'il maîtrise cette coordination oculo-manuelle, la structuration spatiale et la coordination motrice peuvent se faire.

La structuration spatiale est aussi une étape importante. En effet, pour que l'enfant puisse repérer sa droite et sa gauche et pour qu'il répercute cette distinction dans son comportement, il convient qu'il possède le moyen de vivre dans l'espace. Ce moyen, c'est la structuration spatiale. Une preuve de plus, s'il en faut, de cette influence de la structuration spatiale, est que la distinction avant/arrière, développée dès les premiers déplacements de l'enfant, intervient pour beaucoup dans la latéralisation.

Ce qu'on peut dire en fait, c'est que la latéralisation suit la structuration spatiale de l'enfant. L'enfant commence à repérer des éléments sur son propre corps, puis il projette ses repères à partir de son propre corps et enfin, il organise son espace indépendamment de son corps.

La latéralisation est aussi le fruit d'une éducation, d'un apprentissage. Il est acquis que l'enfant possède, de manière innée, les structures mentales propres au développement d'une latéralité ou l'autre. Mais celles-ci ne suffisent pas. La latéralisation ne se fait que par une combinaison d'éléments moteurs, psychologiques, affectifs et environnementaux.

Nous parlons ici de variables environnementales et affectives car le milieu social de l'enfant et ses relations avec ses parents (et surtout sa mère) sont très importants dans le développement de sa latéralité. Notre société est profondément spécialisée pour les droitiers. Aussi, si l'enfant manifeste très tôt une tendance à la gaucherie, son milieu et ses proches auront aussitôt tendance à le reprendre de manière à ce qu'il se comporte en droitier. Cela peut compromettre gravement la latéralité de l'enfant, handicaper gravement sa motricité, entraver sa perception spatio-temporelle.

Dans le développement scolaire de l'enfant tout est fait pour augmenter les capacités motrices (que ce soit gauche et/ou droite). Ainsi vers 3 - 4 ans, les instituteurs et les institutrices commencent par une éducation sensori-motrice. C'est lors de ces exercices que l'enfant pourra choisir une main ou une autre. C'est aussi l'époque où les responsables de l'enseignement des petits commencent les exercices de pré-écriture et de pré-lecture. Il va sans dire que c'est la meilleure période pour détecter les troubles de la latéralité. Ce n'est qu'après, dans les classes supérieures qu'il apprendra à lire et à écrire. Mais la maîtrise de la latéralité est un prérequis à cet apprentissage.

Signalons que les pédagogues actuels ne cherchent plus à modifier le comportement de latéralité d'un enfant. Au contraire, ils encouragent le choix de l'enfant. Si celui-ci manifeste une tendance à la gaucherie, celle-ci ne sera aucunement contrée. Elle sera, au contraire, renforcée afin que l'enfant puisse acquérir une plus grande maîtrise du geste.

Pour terminer, voyons à l'aide de quels tests les professeurs peuvent tester la latéralité de l'enfant. Ceux-ci sont au nombre de 6 (six) :

- **La distribution de cartes** : ce test montre la main directrice. L'enfant doit distribuer des cartes d'une main, puis de l'autre. L'épreuve étant chronométrée, on peut voir que l'enfant est plus à l'aise avec une main qu'avec l'autre.
- **La diadococinésie** : c'est le mouvement des "marionnettes" (c'est-à-dire un mouvement rapide et alterné de prosupination radiocubitale). Ici, c'est la souplesse et la rapidité qui est notée. Evidemment, c'est un geste qui ne pose pas de problèmes pour un adulte. Mais il peut en poser pour un petit enfant.

- **La marelle** : il s'agit de pousser un objet avec le pied en se déplaçant sur une ligne tracée sur le sol. On note ici le pied choisi pour pousser l'objet.
- **Le "shooting"** : c'est l'habilité de l'enfant à frapper une balle du pied. On note la précision du shoot et le pied choisi.
- **Le "sighting"** : c'est ce test qui détermine la direction oculaire, c'est-à-dire l'oeil qui dirige la vision globale (on a toujours un oeil plus fort que l'autre.). Le test consiste à faire regarder l'enfant au travers d'une feuille de carton percée d'un trou. Selon l'oeil avec lequel l'enfant regarde, on peut dire de quel oeil il voit le mieux et donc quel oeil est celui qui dirige la vision.
- **La visée** : ce test met en jeu plusieurs éléments moteurs. Mais il sert surtout à déterminer l'oeil directeur. Les éléments moteurs sont : l'oeil directeur, l'acuité visuelle et la "motricité palpébrale" (c'est-à-dire la facilité à fermer un oeil.). Ce dernier test permet de savoir en fait si la dominance oculaire est stable ou si elle dépend de la main qui tient l'objet à lancer.

D'autres tests permettent en fait de déterminer la latéralité de l'enfant. Ce sont plus des observations de l'enfant que des tests proprement dits. Ce sont en général tous les jeux que l'on fait faire à l'enfant : sauter à cloche-pied, "faire semblant",...

La latéralité est très importante chez l'enfant : son évolution entière est basée sur ce concept. La latéralité influence la façon qu'a l'enfant de se voir lui-même, la perception de son propre corps et de son propre axe de symétrie. Elle est aussi importante dans le développement de la structuration spatiale dont nous parlerons ci-après. En effet, l'enfant perçoit son environnement par rapport à l'axe de son corps.

2.3.1. Les troubles de la latéralité

A chaque étape de son développement, l'enfant peut être arrêté. Soit par une maladie, soit par un accident ou encore par une pression sociale. Il convient de repérer très tôt ces désordres de manière à pouvoir y remédier le plus vite possible.

Dans les troubles de la latéralité, nous avons déjà cité la dextralité fausse et la sénestralité fausse. Pour rappel, il s'agissait d'un trouble atteignant les personnes atteintes dans leurs fonctions cérébro-motrices. Ainsi, si l'hémisphère droit est abîmé à la suite d'un accident ou si le bras (ou la main) gauche est paralysée, c'est, dans le premier cas, l'hémisphère gauche qui doit prendre le relais et la personne peut alors ressentir des troubles à utiliser la main droite. Dans le deuxième cas, l'hémisphère droit du cerveau est toujours actif mais il ne peut plus utiliser la main qui lui correspond. Il y a donc des troubles à utiliser la main gauche en lieu et place de la main droite. Ces troubles peuvent être réduits par l'habitude et/ou la rééducation.

Il y a aussi des personnes qui ne peuvent pas coordonner leurs gestes. Les enfants handicapés moteurs entrent dans cette catégorie. Suite à un dysfonctionnement du cerveau dans ses parties motrices, la latéralité n'a pas pu se faire. L'évolution de ces personnes s'est donc trouvée arrêtée à un certain stade. Généralement ces personnes ne peuvent ni lire ni écrire, leur structuration spatiale est déficiente ainsi que leur orientation temporelle. Ils ont une psychomotricité fine très endommagée rendant pénible le moindre mouvement. Malheureusement, ce handicap moteur se double bien souvent d'un handicap mental. C'est évidemment un gros problème pour les thérapeutes car ils doivent repartir quasi à zéro.

Pour être complet, nous devons citer 3 grandes causes de troubles de la latéralité :

- Les troubles peuvent avoir une origine motrice ou neurologique : c'est le cas des enfants qui sont droitiers de la main mais gauchers du pied.
- Les troubles peuvent avoir une origine sociale : c'est le cas de beaucoup d'enfants qui ne parviennent pas à s'adapter à un environnement fait pour les droitiers. C'est aussi le cas d'enfants que l'on force à utiliser la main droite alors qu'ils sont gauchers, ou enfin, le cas d'enfants imitant un parent qui ne possède pas la même latéralité.
- Les troubles peuvent enfin avoir une origine psychologique : nous avons insisté sur le côté affectif du développement de la latéralité. Mais le côté affectif ne peut être mis seul en cause. L'enfant peut aussi avoir eu des difficultés à établir son schéma corporel.

On peut, bien évidemment, repérer ces troubles par des symptômes. Ceux-ci sont au nombre de 3 : le premier est une hésitation lorsque l'enfant doit choisir une main pour accomplir un geste spécifique. Le second est observé lorsqu'on demande à l'enfant d'exécuter des exercices de précision et/ou des exercices de force. Selon le cas, on remarque alors qu'il se servira tantôt d'une main, tantôt de l'autre. Le troisième est un manque d'homogénéité dans la latéralité : l'enfant choisit la bonne main ou le bon pied (c'est-à-dire le dominant) mais il n'est jamais sûr de savoir où est le côté droit et où est le côté gauche.

Les conséquences sont multiples : l'enfant a des difficultés à reconnaître sa gauche et sa droite, il n'acquiert pas de sens graphique, il forme ses lettres et ses chiffres en "miroir", sa discrimination visuelle est perturbée,...

Heureusement, pour les enfants 'normaux' comme pour les enfants handicapés, il existe des moyens de rééducation. Ces moyens sont des ensembles d'exercices qui s'adaptent plus ou moins bien à un cas ou à un autre. Citons quelques-uns de ces exercices :

- des exercices de perception du côté dominant ou du membre dominant.
- des exercices de perception de l'axe symétrique du corps : cela se fait généralement par des jeux de miroir (l'éducateur fait un geste et l'enfant doit refaire le même).
- des exercices d'assemblage de pantins.
- des exercices de reconnaissance gauche-droite,...

2.3.2. L'ordinateur comme aide...

L'ordinateur peut s'avérer d'un grand secours pour les éducateurs et les thérapeutes. En effet, la plupart des exercices de perception de l'axe du corps, d'assemblages de pantins, de reconnaissance gauche-droite,... peuvent se faire sur l'ordinateur. Il existe beaucoup de programmes qui permettent ce genre d'exercices. Nous pensons notamment à "Corps Humain", un logiciel développé aux Facultés, dans le cadre d'un autre mémoire. Ce programme proposait des exercices de reconnaissance des parties du corps et des exercices de puzzle basé sur le corps humain. Dans un autre genre, "Auteur!" permet, grâce à ses icônes, ses animations, ses sons,... de faire des exercices de reconnaissance gauche-droite,... Il existe encore d'autres logiciels, commerciaux ceux-ci, qui reprennent ce genre d'exercices. Nous en parlerons à la fin de ce chapitre.

2.4. La structuration spatiale

La structuration spatiale est : "l'orientation, la structuration du monde extérieur se rapportant d'abord au moi référentiel, puis à d'autres objets ou personnes en statique ou en mouvement.". Cette définition de J.M. Tasset résume assez bien les différentes étapes nécessaires à l'acquisition de la structuration spatiale.

Nous avons vu, quand nous avons défini la latéralité, que celle-ci se basait sur le schéma corporel. Ce schéma corporel correspond à la prise de conscience, par l'enfant, de son corps, des possibilités motrices de celui-ci ainsi que des possibilités d'action et d'expression.

La structuration spatiale se base sur ces concepts et sur ce schéma corporel. En effet, par la structuration spatiale, l'enfant prend conscience de son corps, non pas en tant qu'entité indépendante, mais comme faisant partie d'un environnement. L'enfant acquiert alors la notion de la place qu'il occupe et de l'orientation qu'il peut avoir par rapport à des personnes et/ou des objets.

En plus de cela, l'enfant acquiert la faculté de voir la situation des choses entre elles, puis la possibilité de déplacer, de faire bouger, d'organiser ces choses entre elles.

La structuration spatiale est très importante. L'enfant est toujours confronté à un environnement. Celui-ci constitue un espace bien précis. Dans cet espace, l'enfant doit :

- se situer lui-même (il est assis *sur* une chaise, *devant* une table...),
- situer les objets entre eux (la gomme est *dans* le plumier, le plumier est *sur* la table),
- organiser les objets entre eux en fonction de l'espace dont il dispose (le soleil est *au-dessus de* la feuille, le chien est *sur* la pelouse,...).

Comme nous le voyons, la structuration spatiale est très importante. Elle participe à tous les événements de notre vie. Ainsi, moi qui écris, je suis devant l'ordinateur et derrière moi le téléphone sonne... Toute notre vie durant, ces notions d'orientation spatiale nous poursuivent. Elles ne sont d'ailleurs pas dissociables des notions de corps et de temps. Ces trois notions, corps - espace - temps, ne peuvent pas être prises séparément sans opérer une limite à un aspect bien précis de la réalité, et si nous les séparons ici, c'est dans un souci de clarté. Dans la réalité, toutefois, ces trois éléments ne sont jamais séparés. Lorsqu'on veut faire acquérir une notion spatiale à un enfant, nous devons prendre en compte ses possibilités, ses connaissances corporelles, son rythme, le temps dont il dispose,...

Pour vérifier que les notions de spatialité sont bien acquises, on fait faire une série de choses en apparence banales à l'enfant. Dans un premier temps, par exemple, on le fait évoluer dans son environnement, dans son espace habituel : "Va dans ta chambre,...", "Porte ce papier sur mon bureau,...". Dans un deuxième temps, on l'invite à se situer lui-même, puis à situer des objets. Sa connaissance des termes spatiaux est vérifiée et éventuellement affinée. Dans un troisième temps, on demandera à l'enfant de trier des objets par ordre de forme et/ou de grandeur.

Une fois qu'il maîtrise bien tous les termes spatiaux, les exercices deviennent plus complexes. C'est l'époque de l'apprentissage de l'orientation : le fait qu'il soit à côté de sa chaise ne suffit plus, il faut qu'il soit à côté de sa chaise ET orienté vers l'avant. C'est pendant cette période que les mots : "Aller vers...", "Tourner à...", "Avant", "Arrière", "Gauche", "Droite", "Haut", "Bas",... prennent toute leur signification.

Quand il a appris à s'orienter lui-même, il apprend à reconnaître des choses orientées d'une certaine façon, puis à orienter ces choses. Une fois cet apprentissage terminé, l'enfant doit acquérir le sens graphique. Quand on voit tout ce qu'il doit apprendre, on voit l'importance de la structuration spatiale dans le développement de l'enfant.

Les apprentissages ci-dessus concernaient l'orientation spatiale. Voyons maintenant ce qui concerne l'organisation spatiale.

Organiser, c'est combiner, disposer,... Nous allons donc, pour cet apprentissage, combiner les différents éléments appris jusqu'ici. Les exercices d'organisation spatiale sont des combinaisons de diverses situations, de diverses orientations. Ces exercices mettent en oeuvre les mêmes éléments : d'abord une organisation par rapport à l'enfant lui-même, puis une organisation entre les objets dont il dispose. L'enfant devra choisir ses propres points de repères et il devra les placer selon diverses orientations pour arriver à la solution de l'exercice. Ces exercices peuvent être variés : guider un camarade qui a les yeux bandés, assemblage de formes pour reconstituer une silhouette, exercice de symétrie,...

Dans cette troisième phase d'apprentissage, l'enfant est seul pour poser les choix. Il n'a plus quelqu'un pour lui dire ce qu'il convient de faire ou de ne pas faire. Il doit mettre en pratique du mieux qu'il peut toutes les notions qu'il a apprises jusqu'ici.

Il y a une notion supplémentaire à la notion de structuration spatiale. C'est la notion de compréhension des relations spatiales. C'est une étape du développement de la structuration spatiale très différente des trois autres parce qu'elle fait intervenir la capacité de raisonnement de l'enfant. Par exemple, lorsqu'on lui présente un exercice de classement de cercles par ordre de grandeur, il doit faire deux abstractions : il doit voir ce qui ne change pas (i.e. le cercle) et ce qui change (i.e. la taille). De plus, il doit pouvoir s'imaginer les éléments manquants entre deux cercles consécutifs. C'est de loin l'étape la plus difficile.

2.4.1. Les troubles de la structuration spatiale

Comme pour les troubles de la latéralité, les troubles de la structuration spatiale sont de différents ordres :

- ils peuvent être dûs à une mauvaise intégration du schéma corporel.
- ils peuvent être dûs à des troubles de la latéralité.
- Ils peuvent avoir pour cause un manque de manipulation : cela veut dire que l'enfant est empêché de créer du désordre autour de lui, on fait pour lui tout ce qui semble difficile. Il ne peut donc manipuler qu'un matériel très limité.
- Ils peuvent, enfin, avoir une origine psychologique :
L'enfant est arrêté dans ses expériences. Il ne peut, par conséquent, occuper l'espace dont il dispose et dont il a besoin. Par exemple, on l'empêche de monter son train électrique dans le salon.
L'enfant distingue mal le rêve et la réalité. Il imagine des tas de choses à partir d'une maison, d'un dessin,...
Son raisonnement est instable : ce qui monte peut aussi descendre, ce qui est à gauche peut être transposé à droite, tout ceci n'a que peu d'importance pour lui.

Les symptômes permettant de détecter ces troubles de la structuration spatiale sont au nombre de 7. Nous allons les détailler afin de bien voir à quelles difficultés ils peuvent mener l'enfant.

2.4.1.a. Premier symptôme : l'enfant ignore les termes spatiaux

Comme il ne connaît pas les termes se rapportant à l'espace, il éprouve des difficultés quand, par exemple, on lui dit de poser son cartable à côté de l'armoire ou de se placer entre une telle et un tel. Dans le premier cas, il déposera son cartable au hasard (par exemple devant l'armoire) et ce sera un hasard s'il pose son cartable au bon endroit. Dans le deuxième cas, il se retournera vers ses compagnons de jeu et ne saura pas où se placer.

Cela a comme conséquences que l'enfant ne participe pas aux jeux des enfants de son âge car il doit toujours trouver un exemple pour pouvoir l'imiter. Il trouve difficilement ses affaires. Il ne peut que difficilement pratiquer les exercices de classement par ordre de grandeur. Sa discrimination visuelle est très mauvaise. Enfin, il est incapable de s'orienter.

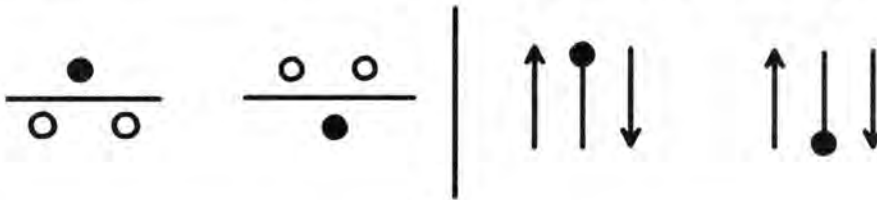
Il existe différents moyens de rééduquer l'enfant. Cela se fait principalement par le jeu et/ou par des exercices (Cf. infra : Les moyens de rééducation).

2.4.1.b. Deuxième symptôme : l'enfant connaît les termes spatiaux mais perçoit mal les positions

Cette mauvaise perception peut aller dans deux sens : le sens haut-bas et le sens gauche-droite. La mauvaise perception haut-bas se traduit par le fait que l'enfant confond des lettres telles que n et u, b et p ou des chiffres tels que 6 et 9. Dans les exercices de discrimination visuelle, les deux signes suivants sont, pour lui, identiques :



Certains enfants savent très bien reconnaître ce qui est en haut et ce qui est en bas, mais à la seule condition que ces notions ne soient pas mêlées à d'autres notions. Ainsi, dans les deux exercices suivants, l'enfant ne parviendra pas à dire où se situe le point noir :



Ici aussi, les moyens de rééducation existent. Nous en parlerons plus loin.

La mauvaise perception gauche-droite, est traduite par la confusion des lettres b et d, p et q, les mots no et on, les chiffres 21 et 12,... L'enfant n'a pas de sens graphique. En calcul, il ne fait pas la différence entre les dizaines et les unités,... En français, que l'adjectif soit placé à gauche ou à droite du nom n'a pas pour lui une importance capitale puisque pour lui, les deux phrases sont identiques. Enfin, pour la discrimination visuelle, les mêmes remarques (transposées) que pour le sens haut-bas peuvent être faites

2.4.1.c. Troisième symptôme : l'enfant perçoit bien l'espace qui l'entoure mais s'oriente difficilement

Dans le cas d'une difficulté d'orientation, plusieurs facteurs sont mis en jeu. Un enfant qui s'oriente difficilement n'est pas capable de s'orienter et de trouver sa place dans un jeu collectif ou dans un jeu d'équipe. Quand ses points de repère changent, c'est son monde qui s'écroule : il ne s'y retrouve plus.

Encore une fois les conséquences sont multiples. Par exemple, lorsqu'un de ses camarades de jeu le poursuit, il court droit devant lui. S'il change de direction, ses repères changent et il est perdu. Si le rang dans lequel il doit prendre place n'est pas à sa place habituelle, il ne parvient pas à y retrouver sa place.

Les enfants perturbés au niveau de l'orientation spatiale ne sont jamais sûrs d'eux. Ils craignent les nouveautés, les grands espaces (à cause du manque de repères visibles). Les jeux de discrimination visuelle ne leur posent pas trop de problèmes sauf s'il y a plus d'un élément qui change... Ils ne parviennent pas à retrouver la place ordinale, même s'ils ont un bon sens graphique. Les exercices d'organisation spatiale tournent souvent au cauchemar... (pour ces enfants).

Au point de vue scolaire, l'enfant parvient à bien lire pour autant qu'il ne doive pas changer de ligne. S'il doit changer de ligne, il ne sait plus où il doit reprendre. En calcul, l'enfant a du mal à comprendre la notion de reports. Il est désorienté (un peu plus) si, dans une suite d'addition, il y a une soustraction.

La rééducation se fait toujours au moyen de jeux divers et d'exercices (Cf. infra)

2.4.1.d. Quatrième symptôme : l'enfant s'oriente bien mais n'a pas de mémoire spatiale

La mémoire spatiale est très importante : c'est elle qui permet de retenir les mots d'une phrase dictée et de les remettre dans l'ordre. Aussi, un enfant qui a des problèmes avec sa mémoire spatiale, parviendra sans peine à recopier une phrase, mais il sera bien en peine lorsque cette phrase lui sera dictée.

A ce stade, bien que l'enfant n'ait pas de problèmes de perception haut-bas ou gauche-droite, il lui est difficile de distinguer b et d. Il a simplement oublié à quoi correspondent ces symboles et il utilise donc l'un pour l'autre, sans s'en rendre compte. En calcul, il peut, par exemple, avoir oublié que le signe "+" veut dire "Ajouter". Dans la vie de tous les jours, il oublie, d'une fois à l'autre, de quel côté de la porte se trouve l'interrupteur,...

2.4.1.e. Cinquième symptôme : l'enfant n'a pas d'organisation spatiale

Un enfant qui n'a pas d'organisation spatiale est un enfant qui heurte sans cesse ses compagnons quand il court, il ne sait jamais s'il passe à gauche ou à droite d'un obstacle. Quand il dessine des personnages, le premier prend toute la place disponible et les suivants sont de plus en plus petits. C'est généralement un enfant qui n'a aucun ordre, ni dans sa chambre (au grand désespoir de sa maman), ni dans son cartable (au grand désespoir de son institutrice) où il ne parvient pas à placer tous les livres.

C'est un enfant à qui il faudra plus de temps qu'un autre pour acquérir la notion d'oblique et donc les notions de lecture et d'écriture n'en seront que plus retardées. En calcul, il comprend bien les reports, mais il ne les place pas à la bonne place.

Les moyens de rééducation seront développés plus loin.

2.4.1.f. Sixième symptôme : l'enfant n'acquiert pas la réversibilité et la transposition

La réversibilité et la transposition sont des notions plus tardives (vers 6 ans). Il n'y a donc d'anormalité que vers 8 ans.

La notion de réversibilité permet à l'enfant de comprendre que, par exemple, 6×7 donne le même résultat que 7×6 . La transposition lui permet de comprendre des abstractions telles que : "Le monsieur mange une pomme" et "Il mange une pomme".

Les difficultés de ces enfants proviennent généralement d'un problème de latéralité et de perception gauche-droite.

2.4.1.g. Septième symptôme : l'enfant a des difficultés à comprendre les relations spatiales

Les relations spatiales font partie de la logique mathématique. Un enfant qui ne perçoit pas les liens entre les choses éprouvera beaucoup de difficultés à comprendre que ce qu'il a appris hier peut encore lui servir aujourd'hui.

Ce sont des enfants qui, s'ils ne sont pas pris en charge, auront beaucoup de mal à synthétiser leurs cours (et ce n'est qu'une conséquence parmi d'autres).

2.4.2. Moyens de rééducation

Chaque symptôme possède ses moyens de rééducation propres. Nous n'allons pas passer tous ceux-ci en revue. Simplement, nous donnerons un aperçu de ce qu'il est possible de faire pour aider un enfant ayant des problèmes.

Il existe une large gamme d'exercices. Nous remarquons toutefois une gradation dans ces exercices. Les plus simples repartent de la base dont l'enfant a besoin : les notions de situation, de mouvement,... Les plus compliqués sont destinés à des enfants qui ont des problèmes correspondant aux trois derniers symptômes.

2.4.2.a. Du plus simple au plus complexe

Les exercices de base sont généralement des exercices de connaissance des termes spatiaux. Ainsi, on fait aller l'enfant d'un endroit à l'autre, soit dans un même local, soit dans une infrastructure plus large (l'école par exemple). D'autres exercices permettent aux enfants de jongler avec ces notions. On les fait jongler avec des notions de situation (dedans, dehors, à gauche, loin, près, autour,...), des notions de grandeur (gros, petit, large,...), des notions de position (debout, à genoux,...), des notions de mouvement (lever, avancer,...), des notions de forme (rond, triangle, carré,...), et enfin des notions de quantité (plein, peu, beaucoup, plus que,...).

Des exercices plus élaborés de tris et de progressions sont efficaces quand l'enfant a déjà acquis certaines notions de base. On fait, par exemple, trier des blocs à l'enfant, par ordre de grandeur, de grosseur,... Les dictées de position, les coloriages dirigés, les classements d'image,... permettent à l'enfant de jongler avec toutes les notions spatiales dans un espace à deux dimensions. Les exercices de placement d'une marionnette dans un décor, de perception dans un décor donné,... permettent à l'enfant de jongler avec la notion d'espace à trois dimensions.

Citons aussi les exercices de reproductions et de reconstitutions : ce sont tous les exercices de puzzle et de dessin par symétrie.

Ces exercices sont parfaits pour faire acquérir les notions d'espace et les termes spatiaux aux enfants en difficulté. C'est ce genre d'exercices qui servent pour la rééducation des deux premiers symptômes.

Les exercices d'orientation spatiale, décrits ci-dessous, sont valables pour les symptômes 3 et 4, à savoir, une mauvaise orientation de l'enfant et un manque de mémoire spatiale. Ce sont des exercices destinés à faire acquérir les notions de file, de rang,... aux enfants. Ce sont des exercices qui se font en file ou en rang. Par exemple, les enfants sont en file, les premiers doivent faire passer une balle entre ses jambes, le second doit la faire passer par dessus sa tête, et ainsi de suite,... Ou, autre exemple, les enfants sont répartis en deux rangs qui se font face et un des deux rangs doit imiter ce que fait l'autre.

Une fois ces notions bien acquises, on passe à des exercices plus compliqués : ce sont des exercices d'orientation spatiale proprement dits. Pour se faire, on peut disposer les enfants dans un certain ordre. A un certain signal, ils doivent se disperser et au second signal, ils doivent revenir à leur place le plus vite possible. Il existe beaucoup de variantes à ce genre d'exercices et il est important que les éducateurs et instituteurs varient le plus possible afin de ne pas lasser les enfants.

Un autre genre d'exercice est l'exercice de mémorisation d'un espace créé : On crée un espace et on demande aux enfants de poser un certain nombre de gestes autour et dans cet espace. Il va de soi que la première fois, cette espace est repérable visuellement. Ensuite, une fois que l'enfant a pu mémoriser cet espace, on lui enlève le repère visuel et on lui demande de refaire les mêmes gestes, sans le repère.

D'autres exercices permettent de renforcer une notion à peine apprise : l'éducateur peut dicter un parcours à suivre dans une salle, on peut dessiner des ronds de couleurs sur le sol et demander à l'enfant de suivre un certain chemin (Attention toutefois, ici, il intervient des notions de rythme pas toujours maîtrisées). D'autres encore permettent de pouvoir retrouver une orientation même si les points de repère changent.

Citons aussi tous les exercices de discrimination visuelle qui font se développer l'orientation et le sens graphique chez l'enfant :

- ♦ Parmi une série de dessins, pouvoir dire lesquels sont identiques,
- ♦ Colorier tous les dessins d'une certaine forme dans une même couleur,
- ♦ Compléter ce qui manque,
- ♦ Colorier les jumeaux,
- ♦ Connaître la place ordinale,
- ♦ Placer et orienter des objets,
- ♦ Reproduire un modèle,...

Il existe aussi une multitude d'exercices de transposition. Ce sont, pour la plupart, des exercices écrits et/ou dessinés. Par exemple, l'enfant doit reproduire un dessin en inversant les couleurs. Ce type d'exercice est très pratique pour les enfants souffrant du 6^e symptôme : la non- perception de la réversibilité et de la transposition.

Les exercices d'organisation spatiale sont d'un autre ordre : ce sont pour la plupart des jeux où l'enfant doit se situer, se repérer,... Ce sont aussi des jeux où, par exemple, on demande à l'enfant de marcher en faisant toujours des pas de même longueur,... Les jeux de cache-cache, ou les jeux où les enfants doivent indiquer un chemin sont également recommandés : ils permettent à l'enfant de se repérer les yeux ouverts. Par exemple, on se rend en un endroit précis et au retour, on demande à l'enfant de guider le groupe.

Certains exercices peuvent combiner plusieurs types : c'est le cas des exercices où l'enfant doit mémoriser un trajet entre divers obstacles puis l'effectuer avec les yeux bandés,...

Ceci conclut les moyens (jeux et exercices) de rééducation des déficits de la structuration spatiale.

2.4.2.b. L'ordinateur et la structuration spatiale....

Beaucoup d'exercices de rééducation peuvent être faits sur ordinateur. Si on excepte les exercices devant nécessairement se passer à l'extérieur (parce qu'ils font intervenir la disposition des bâtiments, ou parce qu'ils demandent des objets non informatisables (un ballon,...)), il y a donc une foule d'exercices que l'on peut réaliser à l'aide de l'ordinateur.

Les exercices qui permettent aux enfants de jongler avec les notions de situation se prêtent bien à l'informatisation : on peut représenter à l'écran différentes situations et demander à l'enfant de les "décrire", on peut demander à l'enfant de placer des éléments dans un décor déjà créé : par exemple, il devra placer un ours *dans* une boîte, et la boîte *sur* un meuble,...

Toutes les notions de situations, de grandeurs et de formes peuvent être mises sous forme d'exercices informatiques. De même, les exercices de tris et de progressions, les dictées de positions, les coloriages dirigés,... peuvent être informatisés.

Les exercices de reproduction et de symétrie sont en revanche plus difficiles à mettre en oeuvre sur un ordinateur. La difficulté majeure vient du fait que la souris est un bien piètre moyen de dessiner sur un ordinateur et malheureusement, c'est le plus répandu.

Les exercices de mémorisation d'un espace créé (Cf. Supra) sont également implémentables sur une machine, ainsi que les exercices de discrimination visuelle, les exercices de transpositions,...

L'ordinateur peut donc servir dans la rééducation des troubles de la structuration spatiale. Et si "Auteur !" n'est pas le plus approprié dans certains cas, il existe une multitude d'autres programmes capables de prendre le relais.

2.5. L'orientation temporelle

L'orientation temporelle est la capacité de se situer dans le temps.

Cette capacité est fonction de la succession des événements : l'enfant apprend le "vocabulaire de base du temps". Les mots "Avant", "Après", "Pendant",... apparaissent dans son vocabulaire.

Cette capacité est aussi fonction de la durée des intervalles de temps. C'est avec ce nouveau concept qu'apparaissent les notions de temps court et long, de cadence rapide et lente, de rythme régulier et irrégulier. Les notions de freinage et d'accélération, d'heure et de minute, de différence entre course et marche apparaissent clairement.

La capacité à s'orienter dans le temps est également fonction de la notion de cycle. L'enfant va apprendre que certaines périodes se renouvellent cycliquement. Ainsi, il perce le mystère des jours de la semaine, des saisons,...

Enfin, l'orientation temporelle est déterminée par la découverte de l'irréversibilité du temps. L'enfant découvre, par l'intermédiaire de son entourage, que le temps ne fait jamais marche arrière. Il comprend (ou on lui fait comprendre), par exemple, que, s'il a 5 ans, 4 ans c'est du passé et qu'il n'aura jamais plus 4 ans. Ici apparaît aussi la notion de vieillissement. Vieillesse des personnes, des objets, des plantes.

Ce sont bien évidemment des concepts très abstraits et il est parfois difficile de les expliquer et de les faire apprendre à un enfant.

Un enfant vit dans le temps. Dès son plus jeune âge, les notions de temps sont présentes mais elles ne sont pas maîtrisées. Ainsi, l'enfant trouve qu'un jeu (événement actif) a passé très vite, ou qu'une sieste (événement passif) a été d'une épouvantable longueur. Dans la tête des enfants, les choses agréables passent toujours trop vite, alors que les choses moins amusantes passent toujours trop lentement.

Ce n'est que vers l'âge de 4 - 5 ans que l'enfant maîtrise les notions d'avant et après. C'est à cet âge qu'il peut reclasser la suite des activités qu'il vit. Cela se remarque très fort dans les histoires que les enfants racontent. Au début, ces histoires sont très confuses : tous les éléments s'y trouvent mais pas dans le bon ordre. Ce n'est que peu à peu que l'enfant parvient à mettre de l'ordre dans ses idées. Ordre étant pris ici au sens strict du mot, c'est-à-dire ordre chronologique.

Les instituteurs(trices) commencent à exercer les enfants à percevoir le temps immédiat dès 4 ans (donc à l'école maternelle). L'éducateur peut, pour se faire, faire dire à l'enfant les gestes qu'il pose, ou cacher différents objets et demander à l'enfant quel objet a été caché en premier.

La difficulté de maîtriser le temps est encore renforcée par le fait qu'il y a deux concepts de temps. Il y a le temps subjectif et le temps objectif.

Le temps subjectif est celui que nous créons par notre propre impression. Ce temps varie selon les individus et selon l'activité du moment. Ainsi, pour l'enfant, un temps de jeu passe plus vite qu'un temps de punition.

Le temps objectif, lui est un concept mathématique. Il est toujours identique. Une heure dure toujours soixante minutes et une minute dure toujours soixante secondes.

Généralement, les éducateurs essaient de développer la notion de temps objectif à partir de points de référence. Ces points de référence sont divers. Ils dépendent de la notion que l'on veut faire passer à l'enfant. Ainsi, l'heure du repas, le jour du bain, le congé du mercredi après-midi,... sont autant de points de repère.

Ce temps objectif est très important dans la vie quotidienne. Et pouvoir s'orienter par rapport à ce temps est encore plus important. En effet la plupart des activités sont réglées en fonction de ce temps : on entre à l'école à 8h30, la récréation est à 10h30, on mange à midi, on sort de l'école à 15h30, on va au lit à 19h30.

La journée de l'adulte est rythmée de la même façon. Seules les heures changent.

Quand l'enfant aura acquis certaines notions telle la durée et la vitesse, il pourra alors pleinement s'organiser. Par cette organisation, il pourra, s'il le souhaite, avoir plus de temps pour ses loisirs.

Prenons un exemple pour illustrer cette organisation :

"L'enfant doit acheter du pain, passer à la pharmacie et terminer ses devoirs avant de pouvoir jouer. S'il réfléchit, il peut faire les courses demandées en revenant de l'école. Il peut passer d'abord à la boulangerie, proche de l'école, puis à la pharmacie qui se trouve sur le chemin du retour. Il s'évitera ainsi un trajet inutile et il aura plus de temps, après ses devoirs pour pouvoir jouer." [Meur,88]

L'orientation temporelle ne se fait pas en une fois. Elle est le résultat d'une longue maturation et elle procède par étapes. Celles-ci sont au nombre de 4 :

- La mise en place de l'ordre et de la succession,
- La découverte des durées et des intervalles,
- L'apprentissage des cycles et du renouvellement cyclique des périodes,
- L'apparition du rythme.

Développons ces 4 points un peu plus en détail :

2.5.1. La mise en place de l'ordre et de la succession

L'enfant doit percevoir et mémoriser des événements qui se passaient avant, qui se passent maintenant, et qui se passeront après. Il doit retenir l'ordre des gestes qu'il pose. Il doit également retenir le premier et le dernier geste. Cet apprentissage peut se faire par le jeu. Il existe en effet une multitude de jeux de gestes, d'imitation,... destinés à éveiller l'enfant à ces notions. Un jeu parmi d'autre est le classement d'images dans un ordre chronologique.

2.5.2. La découverte des durées et des intervalles

L'enfant doit, dans l'étape de découverte des durées et des intervalles, découvrir la différence entre ce qui va vite et ce qui va lentement, entre ce qui est bref et ce qui dure longtemps, entre une heure et une journée. Ici aussi, cette découverte se fait par le jeu. On peut placer l'enfant devant une tâche à accomplir et lui fixer un temps pour le faire. On peut le faire concourir avec ses camarades (courses relais,...).

2.5.3. L'apprentissage des cycles et du renouvellement cyclique des périodes

L'apprentissage des cycles et du renouvellement cyclique des périodes se fait par différenciation des jours de la semaine, des mois de l'année, des saisons,... Pour faire comprendre ces notions à l'enfant, on associe un objet à la période cyclique. Par exemple, une écharpe est associée à l'hiver. Celui-ci devient donc pour l'enfant, "la période où il faut mettre une écharpe". Un autre exemple : on associe une lampe au soir, la piscine au mercredi après-midi,... Ces associations sont multiples et elles seront d'autant mieux retenues que l'association est frappante.

C'est lors de cet apprentissage que l'enfant découvre la question "Quand".

2.5.4. L'apparition du rythme

L'apparition du rythme comprend les notions que l'enfant a découvertes lors des étapes précédentes. Pour que l'enfant acquiert du rythme, l'éducateur, l'instituteur commence par des exercices d'expression corporelle libre sur une musique. L'enfant s'exprime selon son rythme spontané, sans aucune intervention de celui qui dirige l'exercice. Ensuite, on l'invite à suivre les rythmes donnés avec plus de précision : il doit taper dans les mains en rythme avec la musique, il doit marcher en rythme avec la musique.

C'est au cours de cette étape que l'on introduit les rythmes écrits. On demande à l'enfant de reproduire des formes graphiques de manière rythmique. Ainsi, si on lui présente un début de suite comme suit : "OO OO OO", l'enfant devra percevoir le rythme et continuer la série de la même façon.

2.5.5. Les troubles de l'orientation temporelle

Les troubles de l'orientation temporelle peuvent avoir plusieurs origines :

- des origines motrices : liées à un rythme irrégulier de la respiration ou à de mauvaises facultés auditives.
- des origines psychomotrices : nous avons vu que certains exercices de rééducation pour la structuration spatiale faisaient intervenir des éléments de rythmes. Si la structuration spatiale n'est pas faite (ou pas complètement), ou si l'enfant manque d'orientation, l'orientation temporelle sera perturbée.
- des origines psychologiques : l'enfant peut avoir subi un choc affectif (divorce, perte d'un proche,...) ou vivre dans un milieu où les points de repère manquent (par exemple, les repas ne sont pas pris deux jours de suite à la même heure).

On peut repérer les troubles de l'orientation temporelle à l'aide de 4 symptômes que nous détaillons ci-dessous.

2.5.5.a. Premier symptôme : l'enfant est incapable de retrouver l'ordre et la succession des événements

Il est normal qu'un enfant mélange un peu les événements lorsqu'il raconte une histoire. Toutefois, cette tendance à mélanger les éléments doit graduellement disparaître. L'enfant souffrant de ce symptôme ne parvient pas à garder un ordre dans une histoire qu'il raconte. Il ne parvient pas à déterminer ce qui est dernier et ce qui est premier. Il ne situe pas "avant" et "après". Généralement, ce trouble s'accompagne d'une mauvaise organisation dans le sens gauche-droite. L'enfant se trompe dans l'ordre des mots ou n'est pas capable de reconstituer une phrase à partir des mots donnés dans le désordre. En calcul, il ne sait plus quelle opération précède l'autre et cela l'amène souvent à ne pas savoir résoudre des opérations de base.

2.5.5.b. Deuxième symptôme : l'enfant ne perçoit pas les intervalles

L'enfant qui ne perçoit pas les intervalles ne se rend pas compte de ce qui dure longtemps, de ce qui est court, long. Il ne "voit" pas les arrêts. En français, il ne visualise pas les espaces entre les mots : pour lui, que la phrase soit écrite en un mot ou en 5, c'est la même chose. En calcul écrit, il ne perçoit pas les chiffres manquants et ne peut donc aligner correctement les chiffres pour faire une addition ou une soustraction. Quand il fait une lecture collective, il n'est jamais en même temps que les autres.

2.5.5.c. Troisième symptôme : l'enfant n'a pas un rythme régulier

Cette arythmie a des conséquences tant physiques qu'intellectuelles. En effet, sur le plan physique, cette absence de rythme se traduit par une difficulté à la course car il ne parvient pas à réguler ses pas (il fait des pas trop grands et trop petits). Sur le plan intellectuel, il ne parvient pas à lire de façon intelligible ou ne peut associer le geste à la parole lors d'une lecture gestuelle. En calcul, il ne parvient pas à comprendre certains exercices basés sur un rythme de calcul.

2.5.5.d. Quatrième symptôme : l'enfant n'a pas la notion d'heure et ne peut organiser son temps

Comme l'enfant ne peut organiser son temps, il est souvent en retard pour ses travaux. C'est typiquement le cas d'un enfant qui prend bien son temps pour calligraphier sa première ligne puis qui n'a plus le temps pour le reste du texte et qui le bâcle afin d'être "dans les temps". Il n'a pas la capacité de prévoir ses activités. Donc, il vit celles-ci au moment où elles se présentent et ne pense pas qu'il y en a d'autres par après. Il n'a généralement pas le temps de tout faire car il a traîné dans une de ces activités.

2.5.6. Moyens de rééducation

Pour chaque symptôme, il existe des moyens de rééducation. Nous allons citer ici les principaux. Cette liste n'est pas exhaustive : les éducateurs, qui ont de l'imagination à revendre, en trouvent tous les jours de nouveaux. Néanmoins, ils sont assez généraux.

Par rapport au premier symptôme, l'ordre et la succession, les moyens de rééducation sont au nombre de deux. Le premier reprend quelques exercices de spatialité en reprenant les notions de base. Le second consiste en des exercices d'ordre et de succession.

Pour le deuxième symptôme, la non perception des intervalles, il faut présenter à l'enfant les notions de "entre", "contre", "à côté",... Dans un deuxième temps, on peut lui faire faire des exercices de topologie (grilles topologiques), des dictées d'orientation dont on fait varier la longueur, des exercices de durée d'intervalles.

Le troisième symptôme, le manque de rythme régulier, se rééduque par des exercices moteurs de rythmes : marcher au son d'une marche militaire, frapper dans les mains,... Dans une seconde phase, on essaiera des chants mimés, où l'enfant pourra s'exprimer à son rythme. Enfin on terminera par des exercices de rythmes écrits.

Pour le quatrième symptôme, la notion d'heure et d'organisation du temps, la rééducation se fait "montre en main" : on fait faire des exercices de spatialité à l'enfant mais en lui donnant une limite de temps. En fait, ce qu'il faut dans ce cas, c'est faire comprendre à l'enfant qu'il doit s'appliquer, que trop de précipitation nuit, mais qu'il faut consacrer du temps à certaines choses et le reste du temps à d'autres. Il faut faire comprendre à l'enfant qu'il ne doit pas être perfectionniste car sinon, le temps qu'il écrive le titre, ses camarades auront terminé de recopier le texte.

Il faut amener l'enfant, quand il a des exercices à faire, à commencer par les exercices qu'il sait faire et à laisser les plus durs pour après. Il faut l'aider, par tous les moyens, à s'organiser. Attention toutefois à ne pas tout organiser pour lui. Il faut l'aider quand il a plusieurs choses à faire en le conseillant plutôt qu'en lui donnant des ordres.

2.5.7.L'ordinateur et l'orientation temporelle...

La rééducation de l'orientation temporelle se fait surtout par des exercices de spatialité. Surtout pour les trois premiers symptômes. Or nous avons vu que l'ordinateur s'adaptait particulièrement bien aux exercices de spatialité. Pour ce qui est du quatrième symptôme, la rééducation se fait "montre en main". Mais qu'à cela ne tienne, il suffit de poser une limite de temps aux exercices que l'on fait faire à l'enfant sur l'ordinateur. Mais attention dans ce cas de ne pas stresser l'enfant : il faut tenir compte de ses limites et éventuellement commencer avec un temps assez long, quitte à raccourcir ce temps par la suite.

L'ordinateur peut donc aussi servir dans la rééducation des troubles de l'orientation temporelle.

Afin d'être complets quant à la description de la population visée, nous donnerons brièvement les caractéristiques des enfants IMC et des enfants autistes ainsi que leur interaction possible avec l'ordinateur.

2.6. Les Infirmes Moteurs Cérébraux (IMC)

Les enfants infirmes moteurs cérébraux sont des enfants qui ont subi des troubles cérébraux après leur naissance. Une définition plus précise dit que ce sont des enfants qui ont été atteints "d'une anomalie non-évolutive et non-curable des tissus cérébraux, se manifestant entre autres par des troubles moteurs" [BROU,90].

Il existe plusieurs types de troubles moteurs. Les deux plus importants sont la spasticité et l'athétose. Dans la spasticité, les muscles sont raides et hyperexcitables, ce qui provoque une paralysie ou un arrêt des mouvements (les muscles censés se relâcher ne se relâchent pas). Dans l'athétose, c'est le contraire : l'individu a des gestes saccadés, involontaires et incontrôlés. Ces individus ne parviennent pas à coordonner leurs gestes. Leurs actions sont donc souvent maladroites et difficiles.

Comme ces troubles peuvent se situer à n'importe quel endroit du corps, il est difficile de faire une classification dans les problèmes. De plus, ces troubles cérébraux peuvent s'accompagner de troubles intellectuels. Le niveau intellectuel de ces personnes varie entre la débilité profonde et l'intelligence supérieure.

Etant donné ces problèmes d'ordre physique, il convient d'utiliser un matériel informatique adapté : des claviers spéciaux, des techniques de *switches*,... Dans la plupart des cas, pour les IMC, l'ordinateur sert de moyen de communication avec leur entourage. En effet, pour tous ceux qui ne parviennent pas à maîtriser les muscles de leur visage, la parole est quasi impossible. Ils peuvent donc utiliser l'ordinateur pour communiquer avec leur entourage.

L'utilisation de l'ordinateur pour palier à un manque de savoir est, elle, chose beaucoup moins aisée. L'ordinateur, vis-à-vis des IMC sert surtout comme moyen de substitution. Il va permettre, par exemple, à un enfant de jouer à des jeux qu'il ne pourrait pas maîtriser autrement.

Enfin, dans une certaine mesure, l'ordinateur peut servir dans la rééducation des enfants IMC, moyennant l'emploi de dispositifs d'entrée et de sortie adéquats. Bien sûr, il ne faut pas voir l'ordinateur comme l'unique moyen de rééducation, mais il figure parmi ceux que l'on peut utiliser avec plus ou moins de succès, selon que le paramétrage est bien fait ou non.

2.7. Les enfants autistes

Les enfants autistes sont des enfants souffrant d'un problème de communication. Ils ne développent presque pas de relations sociales. Ils ne s'occupent pas du monde extérieur. Ils vivent dans leur monde fait d'habitudes, d'ordre et de routines.

Cependant, il ne faut pas confondre autisme et débilité mentale. Certains autistes sont même très intelligents (certains peuvent calculer la racine carrée d'un nombre à 6 chiffres de tête).

Ces enfants souffrent d'un grave trouble de communication. Ce trouble peut se traduire par un mutisme complet ou par un babil continu. Si vous demandez à un autiste ce qu'il pense de tel film, ou de telle image,... il peut très bien vous répondre par la question que vous lui avez posée.

Si vous dites à un autiste : "Attends deux minutes...", pour lui, au moment où vous lui dites cela, il déclenche un chronomètre dans sa tête et 120 secondes plus tard, il reviendra à la charge.

Ces enfants n'ont pas d'imagination. Ils ne peuvent par conséquent pas inventer un jeu comme le font la plupart des enfants de leur âge. Ils se confinent dans des comportements stéréotypés (le balancement d'avant en arrière en est un) et ne jouent qu'à des jeux appris.

Lorsqu'un enfant autiste a du chagrin, ce qui arrive à tous les enfants, il ne pleure pas. Un enfant autiste ne pleure qu'en cas de douleur. Et si vous lui demandez où il s'est fait mal, il ne désignera pas un endroit sur son corps, mais l'endroit où a eu lieu l'accident.

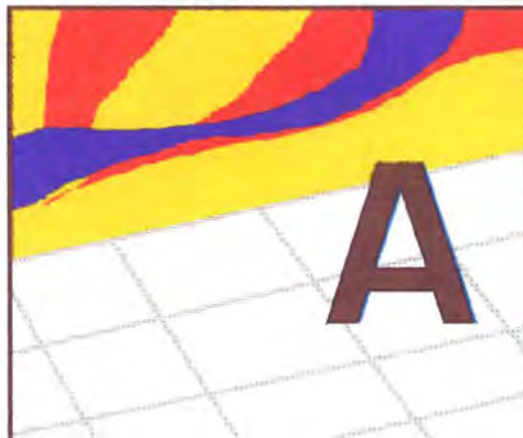
Les enfants autistes sont d'une logique implacable. C'est sans doute pour cela qu'ils aiment travailler avec l'ordinateur. En effet, par rapport au monde extérieur, totalement imprévisible et donc interdit à ces enfants, l'ordinateur est le seul objet de ce monde qui soit entièrement prévisible. L'ordinateur suit donc au plus près leur schémas de pensée.

Les enfants autistes adorent travailler avec un ordinateur car c'est pour eux un compagnon de jeux tout ce qu'il y a de plus logique. Les mêmes données entraîneront les mêmes résultats. Le mode de fonctionnement d'un ordinateur est fort proche de celui d'un autiste. Certains enfants autistes communiquent avec les autres enfants et avec leurs éducateurs uniquement par le biais de l'ordinateur.

L'ordinateur est donc pour eux non seulement un outil de jeux, mais aussi un outil de communication. Il ne faut donc pas les en priver s'ils le demandent.

2.8. En conclusion de ce chapitre

En conclusion de ce chapitre, nous dirions que de pouvoir travailler pour ces enfants est un privilège et nous oblige à ouvrir les yeux sur des concepts avec lesquels nous n'avons pas l'habitude de traiter. Cela nous pose des questions sur notre métier d'informaticien : tant qu'à moderniser la société à l'aide d'ordinateurs, autant n'oublier personne, autant ne pas laisser tous ces gens derrière nous.



Chapitre 3

Expériences extérieures

Où l'on parle des expériences extérieures qui ont conduit à la réalisation du logiciel dans son état actuel. On cite aussi plusieurs autres logiciels existant dans le domaine de l'éducation. Certains ont été testés, d'autres pas...

3.1.Introduction

Dans ce chapitre, nous allons parler des différentes expériences extérieures que nous avons eues dans le domaine des jeux éducatifs. Nous parlerons aussi des stages, lieux où sont utilisés (et même fabriqués) certains de ces logiciels. Les stages se sont déroulés en deux parties. La première, de courte durée (3 jours), s'est déroulée en Angleterre, au ACE CENTER à Oxford. La seconde s'est déroulée à Bruxelles, en collaboration avec le centre pour personnes handicapées "LA FAMILLE". Ce chapitre abordera donc d'abord l'apport des stages dans cette expérience externe, ensuite, sur base de plusieurs articles, nous ferons un rapide (et non-exhaustif) tour d'horizon du marché des programmes éducatifs.

3.2.Quelques jours en Angleterre

Nous avons passé trois jours en Angleterre. Notre lieu de stage était le ACE CENTER situé à Oxford. Ce centre est en fait une école pour enfants handicapés moteurs. A côté de cette école, ou plutôt intégré à cette école, il y a un centre où travaillent des informaticiens. Ceux-ci mettent au point des programmes qu'ils peuvent tester sur place avant de les diffuser. Il va de soi que tous les programmes élaborés dans ce centre sont des programmes destinés au monde du handicap.

Notre visite au ACE CENTER avait principalement quatre objectifs :

- Savoir ce qu'ils utilisent comme matériel,
- Prendre connaissance de leur(s) méthode(s) de développement,
- Etudier le système de paramétrage de leurs logiciels,
- S'intéresser à leurs projets dans ces domaines.

Pour répondre à ces objectifs, nous avons pu observer plusieurs de leurs réalisations dans le domaine de l'informatique et du handicap. Nous avons pu, en fait, voir et utiliser trois de leurs programmes terminés : ACCESSMATH, STARTWRITE et SWITCH ACCESS TO WINDOWS. Nous reviendrons sur ces trois logiciels dans la suite de ce chapitre.

3.2.1. Qu'utilisent-ils comme matériel ?

Les types de machines avec lesquelles ils travaillent habituellement sont des machines très classiques. Il s'agit principalement de PC's (IBM ou clones) et de MacIntosh's. Ils ont travaillé tout un temps avec des ordinateurs de marque BBC (ordinateur développé par ARCON). Malheureusement, ces machines, au demeurant très développées pour le graphisme et auxquelles on pouvait raccorder un grand nombre de périphériques, ne sont plus fabriquées par ARCON. Elles sont remplacées par des machines RISC dotées d'une architecture 32 bits. Ce sont des machines très rapides, mais plus coûteuses. Le succès remporté par ces ordinateurs va donc en s'amenuisant.

Les plates-formes de développement sont essentiellement, pour le MacIntosh, le SYSTEM 7.X, et pour le PC, dans la mesure du possible, MICROSOFT WINDOWS.

Les langages de développement sont adaptés aux plates-formes : sur MacIntosh, ils utilisent principalement HYPERCARD, et sur PC, sous WINDOWS, ils utilisent le C++ et ses techniques orientées objet.

Du point de vue des périphériques adaptés à ces machines, ils utilisent la souris, le clavier, divers systèmes sonores, des "concept keyboard" (nous reviendrons sur cette notion par la suite) et le contact unique (appelé "Switch").

En fait, les deux derniers systèmes périphériques sont dérivés l'un de l'autre. Le contact unique est un dispositif de bouton que l'on enfonce. C'est donc une méthode très pratique pour rendre l'ordinateur accessible à des personnes ayant des déficiences motrices graves.

Extérieurement, le switch est une boîte munie de deux boutons. Pourquoi appeler cela contact unique s'il y a deux boutons ?

En fait, c'est pour faciliter la manipulation : si les déplacements du curseur dans le programme sont verticaux et horizontaux, il est possible ainsi d'avoir un bouton s'occupant de gérer les déplacements verticaux et l'autre permettant de gérer les déplacements horizontaux. Ainsi, la personne peut faire la différence entre les deux sens. Un programme peut toutefois être manœuvré par un seul bouton : tout est question de paramétrage.

A la question de savoir s'il n'y avait pas des cas où il fallait plus de deux boutons, il nous a été répondu que dans ce cas, on utilisait préférentiellement un "concept keyboard". Un "concept keyboard" est un clavier qui comporte un nombre limité de touches. Chacune de celles-ci ayant une fonction spéciale. Ce sont des claviers généralement plus résistants que les claviers classiques, et ce pour des raisons évidentes (risque de mouvements brusques).

Pour revenir au contact unique, intérieurement, il n'y a rien de sorcier dans ce mécanisme. En effet, il s'agit simplement d'une souris sans la bille et sans le boîtier. C'est une solution intelligente car elle permet l'utilisation de tous les pilotes et de tous les programmes de gestion de la souris par WINDOWS. Un switch est donc une souris dont on ne prend pas les déplacements en compte.

Nous développerons la méthode d'utilisation du switch lorsque nous parlerons du programme SWITCH ACCESS TO WINDOWS.

3.2.2. Quelle est leur méthode de développement ?

Les techniciens informaticiens du ACE CENTER utilisent une technique classique dans le développement de ce genre de logiciel : l'évaluation par prototypage.

La première étape du développement est la rédaction d'un document de spécifications qui, dicit M. Colven (informaticien au ACE CENTER) est déjà presque un mode d'emploi du futur programme. C'est ce document qui constitue le cahier des charges de l'application à réaliser.

La méthode suivie par la suite est une méthode très heuristique... Dès qu'une partie du logiciel fonctionne, elle est testée dans l'école du centre (Ormerod School) et dans les autres centres intéressés au projet. Les rapports, remarques, critiques et suggestions diverses sont, dans la mesure du possible, pris en compte dans la suite du développement.

Le processus se poursuit jusqu'à l'obtention d'un produit qui satisfait un maximum de personnes.

3.2.3. Quels sont leurs projets ?

Le ACE CENTER se concentre actuellement sur deux projets :

- le premier est de faire en sorte que leur programme phare, SWITCH ACCESS TO WINDOWS, fonctionne le mieux possible. C'est-à-dire qu'il soit accessible à tous et qu'il soit le plus fiable possible. Ils avaient encore quelques petits problèmes d'affichage sur des normes d'écrans différentes (l'affichage d'une boîte à outils, créée sur un écran VGA, sur un écran EGA, laissait à désirer.)
- le deuxième projet est une participation au projet COMSPEC. Ce projet est international et leur collaboration consiste en l'apport de leur expérience dans le domaine du switch. Ce programme était, en septembre, dans sa deuxième phase de spécification et un prototype tournait déjà. COMSPEC est un programme destiné, d'après ce que nous avons compris, à faciliter la communication des personnes handicapées. De nombreux pays collaborent à ce logiciel : l'Angleterre, la Suède, l'Italie,... Il devrait tourner sur une plate-forme PC, le prototypage MacIntosh ayant été abandonné.

3.2.4. Le paramétrage des logiciels

Le paramétrage des logiciels impliquait que nous voyions les logiciels développés et terminés par le ACE CENTER. Nous allons donc décrire ces programmes brièvement. Nous nous attarderons un peu plus sur SWITCH ACCESS TO WINDOWS, ce dernier utilisant une technique particulièrement intéressante.

3.2.4.a. ACCESSMATH...

ACCESSMATH est un programme éducatif destiné aux enfants qui ne peuvent tenir (et donc utiliser) une règle ou un autre instrument correctement. Il permet de tracer des formes géométriques à l'écran (carré, rectangle, cercles,...). Ces formes pourront être imprimées aux dimensions désirées par l'utilisateur.

Par exemple, l'utilisateur veut dessiner un carré de 3,5 cm de côté. Lorsqu'il le dessine à l'écran, ce carré n'aura pas forcément 3.5 cm de côté. Mais, lors de l'impression, le carré aura bel et bien 3,5 cm de côté.

Un reproche que l'on pourrait faire à ce logiciel est qu'il demande beaucoup de précision dans le maniement de la souris. C'est un problème majeur quand on sait les difficultés des enfants qui seront amenés à utiliser ce logiciel. Toutefois, le logiciel peut être paramétré en fonction des difficultés de l'enfant. Par exemple, une option prévoit d'afficher, en fond d'écran, un quadrillage qui reprend les caractéristiques d'une feuille de papier quadrillée. La règle (ou l'instrument choisi) se déplacera alors de carré en carré jusqu'à être conforme aux désirs de l'enfant.

La figure ci-dessous est une reproduction de l'écran présenté à l'enfant.

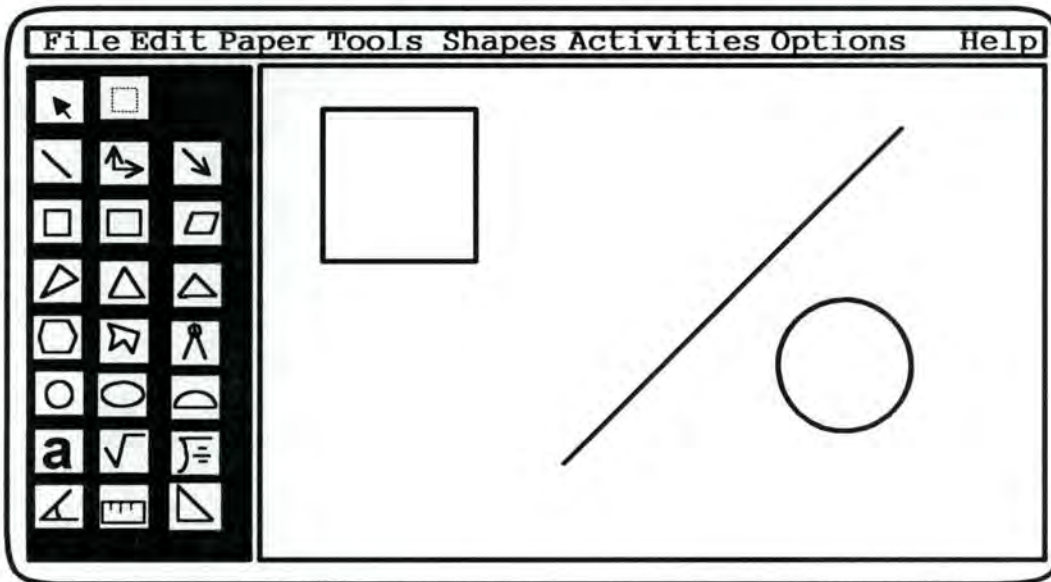


Figure 3.1 : L'écran principal de ACCESSMATH

Ce programme tourne pour l'instant sous DOS. Mais une version Windows est prévue, ainsi qu'une version Archimède (machine RISC).

Les enfants qui utilisent ce programme n'ont pas l'impression d'être différents des autres enfants car ils peuvent arriver aux mêmes résultats.

Malheureusement, à part l'impression des formes dessinées par l'enfant, il n'y a pas de traces de ce que l'enfant a fait. Il peut recommencer son dessin 237 fois, le professeur ou l'éducateur n'en saura rien, sauf s'il reste derrière l'enfant, ce qui n'est pas le but recherché.

C'est un programme qui peut être utilisé dans l'enseignement normal pour apprendre la géométrie aux enfants.

3.2.4.b. STARTWRITE...

STARTWRITE est un logiciel destiné aux personnes (enfants ou adultes) ayant souffert de traumatismes crâniens graves (blessures graves à la tête, lésions cervicales,...). Ces personnes ne peuvent plus écrire parce qu'elles sont incapables de se souvenir de l'orthographe des mots, ou de savoir quel mot il faut employer dans telle phrase, ou enfin d'associer les mots à ce qu'ils représentent.

Ce programme tourne sur PC (mais à condition que celui-ci soit équipé d'une interface sonore) et sur MacIntosh, via l'environnement HYPERCARD.

Par exemple, pour une personne ne parvenant plus à faire le rapport entre son nom et sa propre personne, un exercice parmi d'autres, sera que le professeur, en préparation de l'exercice, tape le nom de la personne et lui associe une prononciation. L'élève aura alors le choix de recopier son nom lettre par lettre, d'écrire son nom à la dictée de l'ordinateur. Cette dictée peut se faire lettre par lettre ou bien par mots entiers.

Tout est paramétrable dans ce logiciel : depuis le temps de réponse prévu, jusqu'au temps entre deux lettres dictées par l'ordinateur, en passant par le nombre de répétitions du mot,... Tout ce qui est erreur, faute de frappe, temps de réponse,... est placé dans un fichier texte que l'éducateur peut visualiser. Cette trace permet à l'éducateur de suivre les progrès de l'élève, de voir les fautes qui reviennent le plus souvent,...

STARTWRITE peut aussi servir dans l'enseignement normal, pour apprendre l'orthographe aux jeunes enfants. Ce programme est un modèle de ce qui devrait être fait du point de vue paramétrage,...

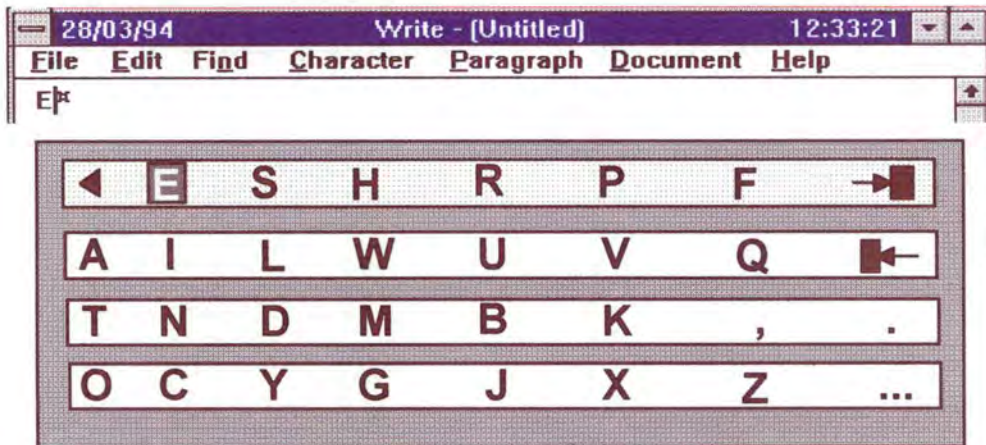
3.2.4.c. SWITCH ACCESS TO WINDOWS (SAW)...

Le programme SWITCH ACCESS TO WINDOWS part d'une bonne idée : rendre accessible tous les logiciels tournant sous WINDOWS et WINDOWS lui-même au personne ne sachant pas se servir d'une souris. Donc, l'idée est d'accéder aux différents programmes (tableurs, traitements de texte,...) à partir de un ou deux switches.

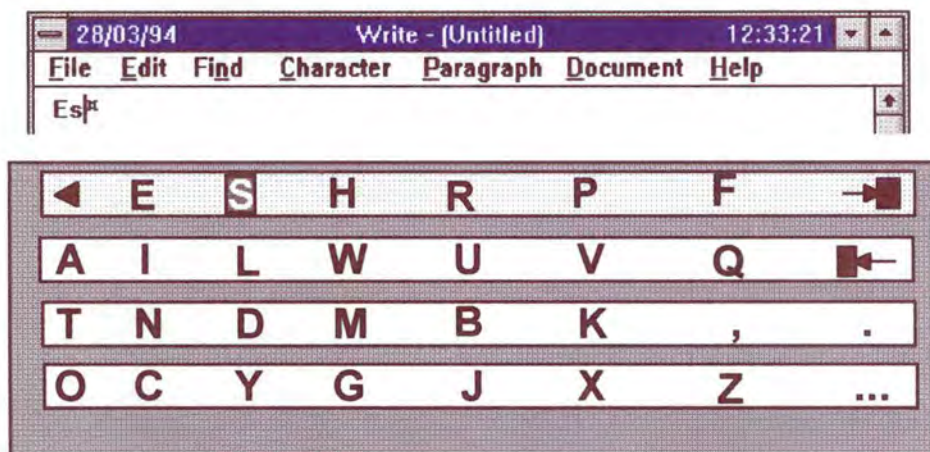
SAW permet de créer une boîte à outils propres à chaque application tournant sous WINDOWS. Les programmes que nous avons pu voir étaient MS-WRITE (traitement de texte fourni en standard avec Windows), WORD FOR WINDOWS (traitement de texte cher à Bill Gates) et KIDPIX (un éditeur graphique pour les enfants).

Tout ce qu'il est possible de faire avec la souris est possible avec le(s) switch(es). Le principe est simple : dans la boîte à outils, des groupes d'outils sont mis en surbrillance chacun à leur tour. Quand le groupe dont on a besoin passe en surbrillance, il faut appuyer sur le bouton du switch. Le déplacement change et la surbrillance passe alors d'outil en outil jusqu'à ce qu'on appuie sur le bouton pour sélectionner l'outil désiré.

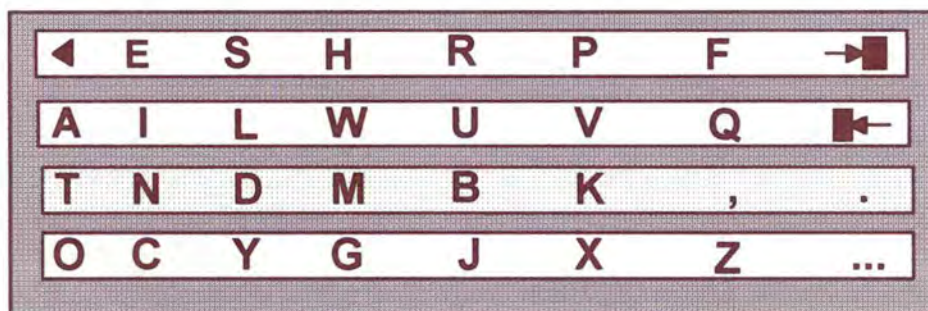
Les figures ci-dessous illustrent l'écriture du mot "est" dans le traitement de texte MS-WRITE dont nous avons parlé ci-dessus. Le but étant bien sûr d'utiliser un balayage pour réaliser cette opération



Dans un premier temps, le "E" est sélectionné pour commencer le mot. Ensuite, on sélectionne le "S" dans le même groupe.



Une fois que le "S" est sélectionné, il faut laisser aller le défilement jusqu'à ce que le groupe contenant le "T" soit sélectionné



A partir de là, le processus reprend à l'intérieur de ce groupe. L'utilisateur doit pousser sur le bouton quand la lettre "T" est en surbrillance.

A partir d'un tel pavé d'outil, il est facile, par l'intermédiaire des "... " en bas à droite, de passer de cette boîte à outil vers une autre contrôlant un autre logiciel. Notons aussi que le contact unique peut être poussé par autre chose que la main : un mouvement de la tête, d'un doigt, de l'oeil,...

Pour une personne entraînée à manipuler ce genre d'interface, cette technique est très rapide. Par exemple, Stephen Hawking, Professeur de physique à l'université de Cambridge, utilise un système de ce genre, et il paraît que sa vitesse est stupéfiante compte tenu de son handicap.

C'est un programme entièrement paramétrable, selon les désirs de celui qui l'emploie. Ainsi, deux personnes utilisant la même application n'emploieront pas les mêmes fonctions de celle-ci. Il est donc important qu'ils puissent définir leur propre boîte à outils. SWITCH ACCESS TO WINDOWS est donc entièrement paramétrable. C'est un programme qui ne demande pas de trace, car son but n'est pas un progrès dans l'utilisation, mais un accès à l'utilisation.

En conclusion de cette partie, nous dirions que ce séjour a été bénéfique malgré sa courte durée. Il nous a montré comment on pouvait adapter un système de switch au programme "Auteur !" et nous a permis de voir comment des professionnels de ce domaine travaillaient.

3.3. Le stage à Bruxelles

Le stage à Bruxelles s'est déroulé au centre "La Famille". Ce centre est un lieu d'accueil pour beaucoup de handicapés de tous les âges. On y rencontre aussi bien des enfants dont les troubles sont plus légers, que des adultes qui ne peuvent quasiment ni bouger, ni s'exprimer. Le but du stage était de permettre une adaptation du programme "Auteur !" à certains problèmes spécifiques dans le domaine du handicap. Nous avons déjà développé ces problèmes dans un chapitre antérieur (Cfr. infra Chapitre 2).

Cette partie du stage a consisté en une suite de réunions avec les responsables (notamment M. Stéphane Lamy, responsable du centre de jour) de l'établissement "La Famille"). Ceux-ci, au cours de ces réunions, ont pu évaluer le logiciel "Auteur !" et ses possibilités. Une fois cette évaluation faite, ils ont exprimé des souhaits, formulé des remarques concernant le logiciel. Ces remarques ont été prises en compte dans la mesure du possible lors de la réalisation des nouveaux outils du programme (Cfr. infra).

De ces réunions, nous avons pu dégager un cahier des charges. C'est ce dernier document qui nous a servi de base pour compléter le programme remis par Luc Vandenabeele. Nous présenterons ce cahier des charges dans un chapitre ultérieur (Cfr. infra Chapitre 4) consacré aux besoins rencontrés.

L'accueil réservé à "Auteur !" dans ce centre pour personnes handicapées a été plutôt bon. Le fait que les éducateurs et thérapeutes du centre utilisaient déjà l'ordinateur dans certaines activités a sûrement facilité les choses.

Deux points de réserve sur le logiciel subsistaient pourtant : premièrement, la méthode à appliquer pour créer des exercices avec "Auteur!" est fort proche d'une méthode "algorithmique". En effet, le créateur d'un exercice doit pouvoir enchaîner des écrans et des actions en fonction de conditions. Cela ressemble très fort au "IF...THEN...ELSE" connus dans presque tous les langages de programmation. Tout le monde n'est pas familier avec ces notions. Il nous a donc été demandé de pouvoir créer, ou au moins de pouvoir donner des indications sur une méthode de création d'exercices. C'est ce à quoi nous nous emploierons dans un chapitre ultérieur (Cfr. infra Chapitre 6).

Le deuxième point de réserve, concernait le mode d'emploi du programme. Il nous a été rapporté que sans le mode d'emploi, on ne pouvait pas utiliser le logiciel car on ne savait pas ce qu'il fallait faire quand on se retrouvait devant l'écran de départ. Pour combler cette lacune, nous avons donc rédigé un mode d'emploi pour le logiciel "Auteur!" Celui-ci se trouve en annexe. Un système d'aide en ligne a aussi été réclamé, mais dans le temps qui nous a été donné, nous n'avons pu mener cette partie du programme à son terme.

3.4. Une évaluation de logiciel éducatifs

3.4.1. Huit programmes testés dans différentes classes

En 1993, le magazine "Computer Magazine" [SCHU,93] publiait un article sur une enquête faite auprès de certaines classes de primaires. Pourquoi une telle enquête ? Peut-être pour marquer l'intérêt des constructeurs de logiciels pour les enfants de 3 à 6 ans. C'est-à-dire, ce que les anglophones appellent les "*preschool children*". De plus en plus d'éditeurs de logiciels ont des produits s'adressant à ce public. Aux Etats-Unis, chaque année, on attribue le "*Best Early Childhood Software*", prix qui récompense le meilleur programme pour les jeunes enfants. Mais, pour pouvoir attribuer ce prix, il faut pouvoir tester les qualités de ces programmes. Pour cela, il n'y a qu'une seule façon de faire : évaluer les logiciels au moment où les intéressés s'en servent, c'est-à-dire par l'observation des jeunes enfants en action.

Il est évident que les ordinateurs entreront, dans un futur proche, de plus en plus tôt dans les salles de classe. Il devient donc nécessaire de pouvoir prévoir les conséquences de cette invasion. C'est dans ce but que l'expérience a été tentée. Huit programmes ont été testés : Mickey's jigsaw puzzle (Disney), Mickey colors and shapes (Disney), Playroom (BroderBund), Adibou (Coktel Vision), Magic (DAInamic Educative Software), Logi (DAInamic Educative Software), Prefix (DAInamic Educative Software), Octobus (DAInamic Educative Software). Ces huit programmes ont été testés dans trois classes différentes : une classe de gardienne, une classe d'enfants handicapés mentaux, et une classe d'enfants autistes.

Les spécialistes ont remarqué, par plusieurs études approfondies, que l'ordinateur était particulièrement bien adapté au public de cet âge. En effet, au risque de nous répéter, nous savons que l'ordinateur est le plus patient des professeurs : il a des réactions immédiates, il s'adapte au niveau de connaissance et au rythme de l'enfant, il ne se fâche jamais en cas d'erreurs répétées.

L'ordinateur a sa place dans une salle de classe, mais il faudra que le professeur veille à ce qu'il n'en prenne pas trop par rapport aux autres activités (rien ne remplace le contact humain). Le professeur devra commencer par établir un vocabulaire de base concernant la machine. C'est en effet primordial pour que les enfants s'y retrouvent. L'utilisation sera réduite, pour les enfants, à un ou deux logiciels ; plus, ils s'y perdraient. L'ordinateur pourra aussi, le cas échéant, servir pour toutes les tâches administratives gérées par le professeur (rédaction d'exercices, lettres aux parents,...). Malheureusement, nous sommes loin de cette situation et ce pour la simple raison que les professeurs ont encore trop souvent tendance à trouver que l'ordinateur est un objet inutile dans leur salle de classe. Certains considèrent même que l'ordinateur ne sert qu'à abrutir et à isoler les enfants. L'expérience menée par "Computer Magazine" tend à prouver le contraire (néanmoins dans les classes maternelles).

En effet, ce qui ressort le plus, dans les discussions avec les professeurs, c'est le fait que les enfants, qui travaillent par groupes de trois, se concertent énormément pour résoudre les problèmes, les énigmes proposées. Chacun ayant un rôle vis-à-vis de la gestion des programmes, ils se respectent mutuellement dans ces rôles. Il a aussi été constaté qu'un objet vertical, un écran par exemple, favorise les échanges entre les enfants.

On remarque aussi deux pôles dans la classe : les enfants qui ont déjà un ordinateur (que ce soit une console ou un autre) chez eux, et ceux qui n'en ont pas. Les premiers ont tendance à moins s'intéresser à l'ordinateur de la classe, et quand ils s'en servent, on constate une bonne maîtrise des outils (souris,...). Les seconds, s'ils n'ont pas la maîtrise de l'outil, l'acquièrent vite. Après quelques minutes, leur coordination "oeil-main" (nécessaire pour faire le rapport entre le mouvement du pointeur de la souris et le mouvement de la main) semble au point.

On constate aussi que l'ordinateur n'est pas, pendant la période d'essai, devenu "le nombril de la classe". Malgré un enthousiasme certain de la part de tous les enfants, on ne peut pas dire qu'ils ne faisaient plus que cela.

On a constaté, au point de vue psychologique, une amélioration de la capacité de concentration, de la coordination "oeil-main", de la prévision des résultats, de la réflexion logique, et de l'endurance. Au point de vue social, d'autres facteurs ont été développés parmi lesquels l'entraide, la concertation, l'expression orale, le respect des tours de rôle,...

Voyons maintenant comment les enfants, que ce soit dans la classe de maternelle ou dans la classe d'handicapés mentaux ou encore dans la classe d'autistes, ont réagi aux programmes proposés.

Dans le monde DISNEY...

La firme Disney, connue pour ses films et ses nombreux personnages, édite des logiciels pour les enfants. Dans les différentes classes, plusieurs de leurs programmes ont été testés. Disney se base sur la connaissance qu'ont les enfants de personnages comme Mickey, Donald, ... et surtout sur la popularité de ceux-ci.

Mickey's Colors and Shapes est un logiciel de reconnaissance de formes géométriques et de couleurs. Il présente tantôt Mickey dans un numéro de magicien, tantôt Pluto dans un numéro de cache-cache. L'enfant choisit des formes et des couleurs à partir du clavier. L'utilisation de ce logiciel doit être soigneusement préparée. Il faut que le professeur prépare le clavier, de sorte que les enfants poussent sur des touches représentant des formes géométriques différentes par les tailles et les couleurs. Notons aussi que l'interface sonore de ces logiciels n'est pas du dernier cri. Le logiciel est en anglais.

Mickey's Jigsaw Puzzle est un programme de puzzle. Il se pilote entièrement à la souris et est paramétrable en difficulté (on peut faire des puzzles comportant entre 4 et 64 morceaux). Quand le puzzle est terminé, l'enfant peut regarder une brève animation. Notons que les puzzles peuvent être imprimés sur papier.

Dans la classe maternelle, c'est ce logiciel qui a eu le plus de succès. Le degré de complexité pouvant être adapté, chacun y trouvait son compte.

Dans les classes d'handicapés mentaux et d'autistes, ces deux logiciels n'ont pas suscité un grand intérêt. Cela réside principalement dans le fait que l'interprétation des formes géométriques laisse fort à désirer chez ces enfants. Or comme le logiciel est basé sur cette interprétation...

Playroom et Adibou...

Playroom est un logiciel britannique comportant 6 sous-logiciels. Le point commun de ces 6 programmes, est la souris Pepper. L'interface est présentée comme un décor de salle de jeu. Un click de la souris sur l'un ou l'autre objet déclenche une réaction immédiate de l'ordinateur. Les 6 logiciels intégrés sont :

Un logiciel de construction : l'enfant construit des pantins ou des animaux en sélectionnant 3 parties du corps.

Un logiciel d'apprentissage de l'heure.

Un programme de reconnaissance de lettres et de mots (pour l'apprentissage de la lecture).

Un jeu de l'oie, entièrement paramétrable quant à sa difficulté. Un reproche toutefois : il nécessite des notions de calcul.

Un jeu de comptage : on montre un nombre d'objets à l'enfant et celui-ci doit indiquer combien il y en a.

Un abécédaire : Un décor apparaît et un alphabet s'affiche au-dessus de l'écran. Lorsque l'enfant clique sur une lettre, un objet apparaît et son nom est prononcé. Cet objet peut alors être placé dans le décor, qui se construit peu à peu.

Adibou reprend pas mal de ces caractéristiques avec en plus la possibilité de suivre tous les progrès de l'enfant.

Dans la classe maternelle, ces programmes ont été appréciés à leur juste valeur. Ces jeux sont automatiquement liés, dans l'esprit des enfants, à un apprentissage. De plus, la simplicité de ces logiciels fait que les enfants peuvent s'en servir sans aucune aide extérieure.

Dans les autres classes, les professeurs n'ont pas utilisé ces logiciels car aucun n'était écrit dans la langue maternelle des enfants. N'excluant pas la possibilité d'instructivité de ces programmes, ils ont décidé d'utiliser les logiciels de DAInamic Educative Software.

Chez DAInamic Educative Software...

Quatre logiciels ont été retenus : Magic, Logi, Prefix et Octobus.

Magic est un logiciel de coloriage amélioré. Il permet de colorier des formes, de faire des puzzles, de faire des exercices de différenciation de couleurs. Élément intéressant : on peut ajouter des images pour les puzzles et les coloriages. Ces images, dans un format très courant (PCX) permettent au logiciel de rester attrayant.

Dans la classe d'autistes, ce programme a été très vite accepté. La possibilité de réglage de la souris offerte par le programme a été jugée "une bonne idée", surtout pour les enfants ayant des problèmes de coordination oculo-manuelle. Un exercice ne permettant pas le coloriage de plusieurs morceaux dans la même couleur a été particulièrement apprécié pour les enfants ayant une tendance trop forte à l'imitation. Un reproche de la part des responsables de cette classe : le programme n'a pas de fin fixée. C'est un gros problème car des enfants autistes se posent toujours trois questions : que faire ? pendant combien de temps le faire ? que faire après ? Dans ce cas, le logiciel ne répond pas à la seconde question.

Dans la classe d'handicapés mentaux, seuls les enfants profondément handicapés ont passé des heures avec les exercices de coloriage. En règle générale, le niveau du programme était trop bas.

Dans la classe maternelle, un concept a été regretté : l'emploi des touches de direction pour la manipulation du programme. Toutefois, une rapide explication des touches et surtout de la touche "ENTER" comble rapidement les lacunes de chacun.

Logi est un logiciel destiné à enseigner les bases du calcul au moyen de blocs. Nous avons tous joué à ces jeux avec les briques lorsque nous étions plus petits. Des briques de différentes tailles permettent à l'enfant d'apprendre à compter. C'est ce principe qui a été repris ici, mais sous forme informatique.

Dans la classe maternelle, le logiciel a été jugé bon pour le service pour son côté apprentissage, mais malheureusement pas pour son côté animation et ludisme. Les animateurs doivent constamment rester dans les parages de la machine pour stimuler les enfants.

Dans la classe d'autistes, les professeurs ont trouvé le logiciel intéressant pour les notions qu'il apporte de gauche, droite, au-dessus, en dessous,...

Dans la classe d'handicapés mentaux, l'accueil réservé au programme a été assez froid. Soit le logiciel n'était pas assez passionnant, soit son niveau était trop bas.

Prefix est lui aussi un logiciel d'apprentissage de la base des mathématiques. L'enfant peut faire correspondre un ensemble de fruits, d'animaux à un chiffre donné. Dans un autre exercice, il devra retirer un certain nombre (donné) de billes d'un ensemble pour les placer sur un terrain de jeu. D'autres scénarios sont prévus, très variés.

Dans la classe maternelle, les enfants avaient tous la capacité d'utiliser ce programmes par eux-mêmes. Peut-être était-il trop facile ?

Dans la classe d'autistes, Les enfants pouvaient travailler seuls grâce à la présentation structurée des exercices. Malheureusement, certains changements de couleurs ont désorienté les enfants, et la représentation des chiffres n'était pas la même, dans le logiciel, que celle qu'ils utilisaient en classe. Cette dernière remarque est très importante : *quelle que soit la classe ou l'on teste un programme, il faut veiller à ce que le programme respecte la méthode d'apprentissage à laquelle les enfants sont habitués.*

Dans la classe d'handicapés mentaux, les enfants les plus jeunes ont éprouvé quelques difficultés avec les scénarios impliquant les séries de nombres. Mais dans l'ensemble, le logiciel a été bien accepté.

Octobus, comme son nom l'indique, est un ensemble de 8 logiciels. Ce sont des jeux de réflexion, d'association qui font appel à des notions aussi variées que la discrimination visuelle, la discrimination auditive, les connaissances linguistiques et/ou mathématiques,...

Dans les classes maternelles et d'handicapés mentaux, le programme a reçu un accueil très chaleureux. Chacun y trouvant son compte.

Dans la classe d'autistes, le programme a été apprécié car il tient compte du temps. En effet, après chaque exercice, le score et le temps sont affichés. Cela pousse ces enfants à essayer de battre leurs propres records. La motivation à l'utilisation est donc très forte.

3.4.2. En complément d'information

Plus récemment, dans un article consacré aux programmes éducatifs, la revue PC HOME ([REES, 94]), a publié une évaluation des jeux et de logiciels éducatifs. Ces logiciels ne sont pas spécialement destinés au domaine du handicap. Toutefois, certaines parties peuvent être utilisées dans ces domaines plus particuliers. L'auteur de l'article, M. Wilf Rees, nous donne une définition de ce que devrait être un programme éducatif (NDLA : cette définition n'a de sens qu'en anglais puisqu'elle est constituée par l'acrostiche du mot EDUCATION. Voici cette définition : "

- *Easy to install and use - manuals should be virtually unnecessary*
- *Dedicated to specific areas or subjects, with explicit objectives in mind*
- *Unbiased and moral, engendering non-discriminatory values*
- *Cheap enough for parents and students to buy for themselves*
- *Accurate and entertaining, to ensure attention holding*
- *Thorough in its conveying of a subject or principle*
- *Intuitive, allowing testing, measuring, monitoring and reporting of performance*
- *Objective, reinforcing accumulative learning patterns*
- *National curriculum dedicated, with Key Stages and levels identified. "*

Tous ces points, nous le pensons, sont assez évidents. Seul le dernier demande, à notre sens une petite explication : Les mots "*National curriculum*" font référence au programme officiel des cours dans les écoles américaines. Aux Etats-Unis, les enfants commencent l'école à 5 ans et y restent au moins jusque 16 ans. Ces années d'école sont divisées en 4 catégories, ce que les américains appellent les *Key Stages*. Chaque *Key Stage* est lui-même divisé en 10 niveaux d'apprentissage, de connaissance. A la fin de chaque *Key Stage*, l'enfant doit atteindre le 10^e niveau de connaissance avant de passer au suivant. C'est ainsi que l'auteur de cet article préconise que les programmes devraient être dédiés à des *Key Stages* particuliers.

Nous allons maintenant parler des programmes qui ont été testés dans cet article. Ces programmes sont au nombre de 27. Bien sûr nous ne les présenterons pas tous, mais vous pourrez trouver, à la fin de ce chapitre, un tableau récapitulatif des 27 programmes testés dans l'article.

Dans le monde EUROPRESS Software

La firme EUROPRESS Software est une firme qui contribue beaucoup à l'extension du marché des programmes éducatifs. Beaucoup de leurs produits sont très impressionnants et particulièrement bien adaptés aux tranches d'âge que nous regardons.

EUROPRESS Software est l'éditeur de "*Fun School*". "*Fun School*" est un logiciel éducatif contenant 6 jeux, chacun possédant trois niveaux de difficultés. "*Fun School*" en est à sa quatrième version. Ce programme existe en "trois tailles" : en dessous de 5 ans, entre 5 et 7 ans et enfin entre 7 et 11 ans. Les jeux proposés sont bien sûr adaptés à l'âge du joueur.

Ce sont des jeux qui couvrent des sujets très vastes : les bases du calcul, de la lecture et de l'écriture, le concept du temps, la gestion d'un budget, l'histoire, la géographie,...

En addition à la série "*Fun School*", il existe un programme qui s'appelle "*Fun School Paint and Create*". Ce programme permet à l'enfant d'apprendre à manipuler le clavier, à faire de la musique et des graphiques,...

Dans le monde ADI

Nous avons déjà parlé du monde ADI avec le programme ADIBOU (Cf. Supra). La firme ADI a commercialisé d'autres programmes. Ainsi, la série *ADI Junior* est une série qui connaît un certain succès. Il en existe quatre versions : deux versions *ADI Junior Reading* (pour les enfants de 4-5 ans et pour les enfants de 6-7 ans) et deux versions *ADI Junior Counting* (l'âge ciblé est le même).

Dans le monde Electronic Arts

Electronic Arts n'est plus une firme que l'on présente. C'est une entreprise qui s'est surtout distinguée pour ses réalisations dans le domaine du jeu vidéo. Elle vient de faire une entrée remarquée dans le domaine du logiciel éducatif. Elle propose trois ensembles : "*Putt Putt joins the parade*", "*Putt Putt goes to the moon*" et "*Putt Putt's Fun Pack*". En plus du logiciel, Electronic Arts propose, dans l'emballage, tout un lot de crayons de couleurs, d'images à colorier,...

Le seul reproche que l'on pourrait faire à ces ensembles de jeux, est qu'ils demandent un gros effort de logique de la part de celui qui y joue. Ainsi, dans un des jeux, il faut aider le personnage à réparer sa fusée pour pouvoir rentrer sur terre. La réparation de la fusée demande des notions de logique assez poussées, notions qui ne sont pas à la portée de tous les enfants pour lesquels ce logiciels est destiné.

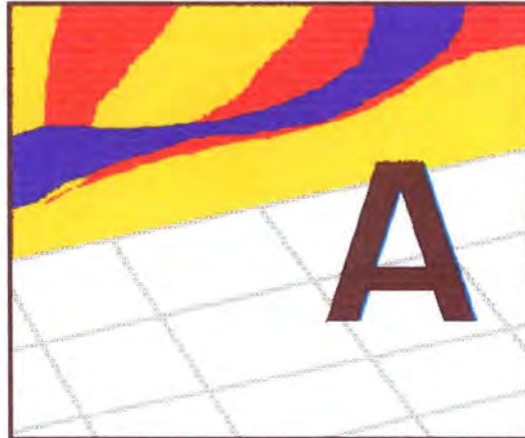
Il y a encore beaucoup d'autres logiciels, mais faire la liste explicative des 27 programmes testés serait trop long et surtout hors de notre propos. C'est pourquoi, nous avons décidé de donner le tableau comparatif des différents programmes testés dans l'article. C'est ce tableau qui figure ci-dessous.

Titre	Sujet	Age	Doc	Présentation	Editeur	Cote globale /6
Putt Putt goes to the moon	General	3/8	3	6	EA	3
Putt Putt joins the parade	General	3/7	3	6	EA	3
Putt Putt's fun pack	General	3/8	4	4	EA	4
Noddy's playtime	General	3+	6	5	JB	5
Noddy's big adventure	General	3+	6	6	JB	6
Fun School 4 under 5	General	>5	5	4	ES	5
Fun school 4 5 to 7	General	5/7	5	4	ES	5
Fun School 4 7 to 11	General	7/11	5	4	ES	5
Fun School Paint and Create	General	5/11	3	4	ES	3
Adventure in Fairyland	General	4/8	3	5	SS	4
Creativity Centre	General	5/10	3	5	SS	4
Upside Town	General	5/7	3	5	SS	4
Mind Games	General	5/10	3	4	SS	4
Fun School Spelling	English	7/11	5	4	ES	4
Storybook Weaver	English	5+	4	5	ES	4
Mutanoid Word Challenge	English	5+	2	3	ES	3
ADI Junior Reading 4/5	English	4/5	5	4	ES	4
ADI Junior Reading 6/7	English	6/7	5	4	ES	4
Spellbound	English	5+	3	4	LS	4
Reading Adventure in....	English	4/7	3	4	SS	4
Henriette's book of spells	English	5+	5	5	LS	5
ADI Junior Counting 4/5	Maths	4/5	5	4	ES	4
ADI Junior Counting 6/7	Maths	6/7	5	4	ES	4
Hooray for Henrietta	Maths	5/11	5	4	LS	5
Fun School Maths	Maths	7/11	5	4	ES	5
Count and Add	Maths	>6	2	4	LS	3
Rainbow Rascal	Graphic	3+	3	4	SS	3

Légende : ES = EUROPRESS ; EA = Electronic Arts ; JB = Jumping Bean ;
LS = Lander Software ; SS = Scops Software.

3.5. En conclusion de ce chapitre

En conclusion de ce chapitre, nous voulons revenir sur le travail que nous avons fourni cette année, et plus spécialement sur la partie que nous avons effectué en collaboration avec le centre "La Famille". Nous avons envie de dire que rien ne remplace le contact avec les utilisateurs. En effet, au départ, nous ne voyions pas comment nous aurions pu implémenter des exercices concernant les problèmes que nous avons évoqués au chapitre 2. Les éducateurs du centre "La Famille", par l'intermédiaire de M. Lamy, nous ont aidé à y voir plus clair. Le chapitre suivant nous servira à illustrer les besoins rencontrés et les solutions que nous avons proposées.



Chapitre 4

Quoi de neuf ?

Où l'on parle des nouveaux outils implémentés à la demande des éducateurs et thérapeutes. Ce chapitre constitue en quelque sorte un cahier des charges du travail que nous avons fait.

4.1. Introduction

Ce quatrième chapitre fait l'objet de la description des besoins des utilisateurs du logiciel. Ceux-ci sont amenés à traiter des problèmes relatifs à ceux que nous avons développés précédemment. Comme nous l'avons signalé à la fin de la troisième partie, ce chapitre est en fait une sorte de cahier des charges. Il reprend les besoins (les demandes spécifiques,...) et les solutions apportées.

4.2. Les besoins rencontrés

Les besoins rencontrés sont nombreux. D'une part parce que certains éléments essentiels manquaient au programme. D'autre part parce que nous prenions en compte des exercices destinés à un autre public et donc les exercices que l'on pouvait développer jusqu'à présent n'étaient plus appropriés à la population avec laquelle on voulait travailler.

Nous divisons les problèmes en trois catégories : les impératifs, nécessaires au bon fonctionnement du programme, les secondaires, apportant un petit plus, et les facultatifs, qu'il serait bon d'avoir.

4.2.1. Les besoins impératifs

Les besoins impératifs étaient au nombre de cinq (5). Ces cinq problèmes sont chacun une amélioration ou un complément du programme originel de Luc Vandenaabeele. Ce sont aussi des morceaux de code que mon prédécesseur n'a pas eu le temps d'écrire et de tester. Malheureusement, ce sont des choses essentielles au bon fonctionnement du programme.

Ces cinq points sont les suivants :

- *Les opérations de sauvetage et de rechargement des exercices créés* : le programme, fourni par Luc Vandenaabeele en septembre 1993, ne comportait pas les instructions de sauvetage et de rechargement des exercices et écrans déjà créés. Evidemment, c'est un handicap grave pour un programme que de ne pas pouvoir sauver le travail que l'on vient de faire. C'est donc avec la plus haute priorité que le sauvetage et le rechargement d'exercices ont été placés dans le cahier des charges.
- *Le détournement des pictogrammes des icônes* : Quand l'utilisateur exécutait un exercice, il devait juste cliquer sur le carré contenant l'icône. C'était une solution acceptable pour certains exercices ou pour certaines personnes. Mais pour d'autres exercices ou d'autres personnes, ça n'allait plus. Ces autres personnes ne voyaient pas le rapport entre leur action (quand il clique dans le carré mais pas sur le dessin) et ce qui se passe après. Pour remédier à cela, il fallait pouvoir détourner les images que l'on place dans les icônes, afin que l'utilisateur doive cliquer *sur* le dessin et pas à côté.
- *Le système de coordonnées à l'écran* : le système de coordonnées à l'écran, comme prévu dans l'action de déplacement d'un objet, n'était pas pratique pour celui qui crée des exercices. En effet, il devait calculer la position de l'objet en pixels. Or un pixel (ou point de l'écran) n'est pas une unité très précise. De plus c'est une unité très abstraite. L'utilisateur ne voit pas ce qu'il fait en disant que l'objet doit aller à

la position <350,267>. Il faut donc trouver un autre moyen d'indiquer les coordonnées écran.

- *Ajouter des moyens d'interaction* : Le programme ne fonctionnait qu'avec le clavier et la souris. Il nous a été demandé d'ajouter le switch et le joystick (si possible). Cela pour faciliter l'accès au programme des enfants atteints de problèmes de motricité.
- *Commenter le code remis par Luc Vandenabeele* : Ce code n'était pas du tout commenté. Il était donc très difficile d' "y rentrer" pour pouvoir le réutiliser.

4.2.2. Les besoins secondaires

Les besoins secondaires sont des besoins que l'on aimerait implémenter sur certains exercices seulement. Ce sont les contrôles de validité, afin de ne pas laisser faire n'importe quoi à l'utilisateur et surtout au créateur d'exercice (actuellement, il peut tout faire) et les analyses de trace, c'est-à-dire, que les actions faites par celui qui utilise les exercices sont répertoriées dans un fichier. Ce fichier, il faut pouvoir l'analyser de manière efficace, afin de pouvoir adapter les exercices à ceux qui les font.

Les autres besoins secondaires sont des besoins purement pratiques : vérifier la compatibilité des écrans, constituer une bibliothèque d'images et de sons, signaler comment on peut ajouter des images et des sons, faire de la navigation d'objet à l'écran,...

4.2.3. Les besoins facultatifs

En fait, ces besoins facultatifs se résument à un seul besoin : l'aide en ligne du programme "Auteur!". Mais cet objectif est assez difficile à réaliser : une aide en ligne doit être soigneusement préparée et pour la préparer soigneusement, il faut du temps. Il faut également du temps pour bien la rédiger et la relire afin de voir si on n'a rien oublié. Bref, l'aide en ligne du programme n'a pas encore pu être faite, mais nous ne désespérons pas de pouvoir la commencer.

4.3. Les solutions proposées

Les solutions que nous avons proposées pour chaque problème sont toutes assez simples. Voyons un peu comment nous nous proposons de répondre aux besoins des utilisateurs :

4.3.1. Les solutions proposées pour les besoins impératifs

4.3.1.a. Le sauvetage et le rechargement

Dans la version de septembre 1993, l'utilisateur avait la possibilité de sauver tout un exercice, ou seulement un écran de l'exercice. Depuis, l'utilisateur, qui n'avait la possibilité que de recharger des exercices entiers, peut également recharger des écrans d'exercices préalablement sauvés. La structure d'un exercice s'adapte spécialement bien à ce genre de sauvegarde. En effet, un exercice est composé d'écrans, un écran est composé d'objets et à un objet est reliée au moins une action.

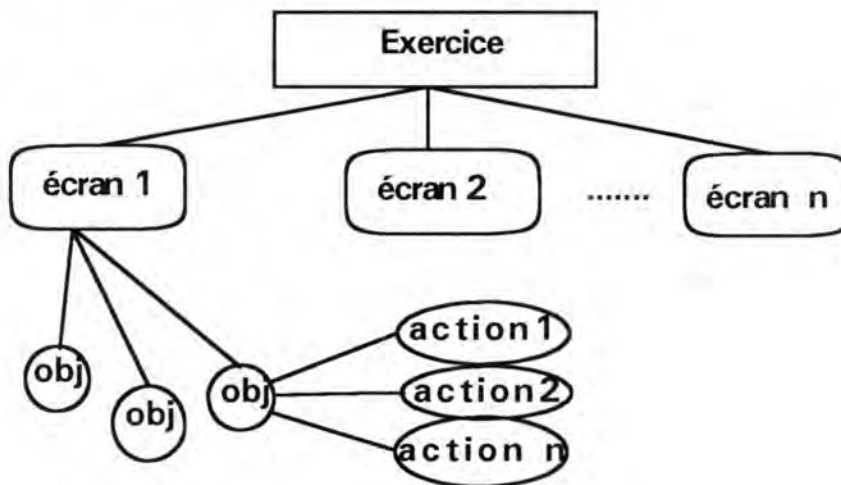


Figure 4.1. : représentation de la structure d'un exercice.

Cette structure nous permet de faire ce que nous nous proposons : rien, en effet, ne nous empêche de couper cette structure ou bon nous semble pour pouvoir sauver un seul écran. Il suffit de le séparer de son exercice. De même pour le rechargement, on peut attacher un écran indépendant à cette structure : il suffit de dire au programme que ce qu'on charge n'est pas un autre exercice, mais un morceau à greffer sur un exercice existant.

Si on veut sauver un exercice entier, on sait quels écrans le constituent. Partant de là, on sait quels objets sont reliés à un écran, et quelles actions sont reliées à un objet. Cela donne la structure de fichier suivante :

Nom du fichier.

Zone destinée aux paramètres du programme.

Nombre d'écrans

Nom du premier écran.

Caractéristiques de l'écran : couleur, actions associées,...

```

Nombre d'objets à l'écran.
    Nom du premier objet.
    Caractéristiques de l'objet : fichier image associé, position,...
    Nombre d'actions associées.
        1° action
        Caractéristiques : fichier son, paramètres,...
        ...
        Fin 1° action
        ....
        n° action
        Caractéristiques
        ...
        Fin n° action
    Fin 1° objet
    .....
Fin 1° écran.
....
Fin de fichier.

```

4.3.1.b. Le système de coordonnées à l'écran

Le positionnement d'un objet à l'écran est une action qui existait déjà. Nous n'avons fait qu'améliorer la façon de paramétrer l'action. Jusqu'en septembre, il fallait entrer les coordonnées à la main. Le désavantage de cette solution était que l'utilisateur devait calculer les coordonnées de l'objet à l'écran. Cela lui rendait le travail plus difficile car, comme nous l'avons déjà exprimé auparavant, l'unité de coordonnées écran est le pixel et c'est une mesure très abstraite (1 pixel sur un écran VGA ne se voit quasi pas !). Nous avons donc imaginé un système permettant à l'utilisateur, non plus d'entrer les coordonnées à la main, mais de cliquer à l'endroit où il souhaite que l'objet apparaisse. Les coordonnées du point choisi sont alors transférées dans les champs qui servaient à définir ces coordonnées.

Résumons donc la situation, lorsque l'utilisateur choisit l'action positionner un objet, et lorsqu'il paramètre cette action, il voit la boîte de dialogue suivante :

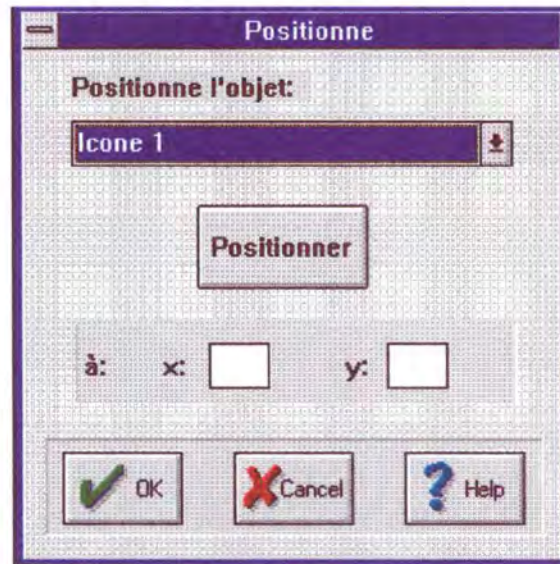


Figure 4.2. : La boîte de dialogue "Positionner un objet"

Le premier champ de cette boîte permet de choisir l'objet que l'on va déplacer. C'est une "drop down combo box" qui contient tous les éléments de l'écran en cours (il ne nous est pas apparu opportun de placer tous les éléments de l'exercice dans cette liste, et ce pour la bonne raison qu'il ne sert à rien de déplacer un objet dans un écran que l'on ne voit pas). Le bouton "positionner" fait apparaître une boîte de dialogue entièrement vide à l'exception de deux boutons. Son fond est blanc et seul le bandeau permet de savoir ce qu'il faut faire. C'est dans cette boîte que l'utilisateur doit cliquer pour désigner la nouvelle position de l'objet. Seul le dernier click de l'utilisateur est pris en compte.

On pourrait, bien sûr, faire un contrôle de la position par rapport au bord de l'écran. En effet, si l'utilisateur clique trop près du bord, l'objet ne sera pas accessible, car il sera affiché en dehors de l'écran. Il faut donc interdire une certaine zone à l'utilisateur. Quand celui-ci a terminé son choix, il clique sur le bouton "OK", et les coordonnées choisies apparaissent dans les deux champs X et Y de la première boîte (celle que nous avons représentée ci-dessus). S'il est satisfait de son travail, il clique sur "OK", sinon, il a toujours la possibilité de recommencer. Notons que les deux derniers champs ne sont pas modifiables par l'utilisateur. S'il veut modifier les coordonnées, il doit réappuyer sur le bouton "positionner".

Notons également qu'une "mini" aide en ligne est prévue : lorsque l'utilisateur positionne le curseur de la souris sur un bouton ou un champ quelconque dans une boîte de dialogue, sur la barre d'état, en dessous de l'écran, devrait s'afficher une phrase qui permettrait à l'utilisateur de savoir à quoi correspond l'action qu'il envisage d'effectuer. Cela diminuerait le risque de manipulation erronée et cela limiterait l'emploi du mode d'emploi du programme. En effet, il n'est pas agréable de se référer à un mode d'emploi à chaque fois que l'on utilise telle ou telle option du programme. La solution de la barre d'état nous apparaît donc comme étant la meilleure solution de remplacement en attendant l'aide en ligne réelle.

4.3.1.c. Le détournage d'objet

Le détournage d'objet est, parmi les points impératifs, celui qui nous a donné le plus de mal. En effet, dans la structure prévue du programme, tous les éléments sont des "child windows" de l'écran, lui-même "child window" de la fenêtre de l'application. Or une "child window" est une fenêtre comme une autre, et une fenêtre, sous Windows, est toujours rectangulaire.

Le problème du détournage se posait donc en les termes suivant : comment peut-on changer la forme d'une fenêtre de manière à ce que celle-ci prenne la forme du dessin que l'on colle à l'icône ? Il va sans dire que nous n'avons pas réussi dans cette voie. Mais, nous avons trouvé une parade à ce problème : Windows nous permet de définir des régions polygonales à partir d'un tableau de points. De plus, Windows nous fournit les instructions nécessaires pour tester si un click souris a eu lieu dans cette région ou en dehors. Nous sommes donc partis sur cette piste.

La première partie de la solution consiste à ajouter un bouton dans la boîte de dialogue de réglage d'une icône. Ce bouton appelle une autre boîte de dialogue. Cette boîte de dialogue contient une série de boutons et une représentation du pictogramme. Le principe est assez simple : à partir de cette boîte de dialogue, l'utilisateur peut définir un contour plus ou moins précis à son icône. Cela se fait en définissant des points tout autour du dessin. Ces points sont repris dans un tableau de points. Ce tableau de points sert par la suite pour créer la région. Le reste du dessin ne sera donc, une fois cette région définie, plus pris en compte lors d'un click en dehors de la région. Mieux encore, le fond de l'image peut être défini transparent. Ce qui a l'avantage de ne plus montrer un carré blanc sur un fond de couleur. En effet, si le fond de l'icône est transparent, lorsqu'on revient à la conception, le fond de l'icône sera de la même couleur que le fond d'écran. Ci-dessous, nous montrons à quoi ressemble cette boîte de dialogue permettant de définir un contour d'image.



Figure 4.3. : Détourage d'un objet.

Par défaut, la région est constituée de quatre points, correspondants aux quatre coins de l'image. Le bouton "Mise à zéro" sert à restaurer la région à ces quatre seuls points. Le bouton "Fond invisible" sert à rendre, comme nous l'avons déjà expliqué, le fond de l'icône de la même couleur que le fond de l'écran. Le bouton "Ajouter un point" permet d'ajouter un point dans le tableau. Le bouton "Oter un point" fait l'opération inverse.

Remarquons que dans cette boîte de dialogue, on ne fait que définir des points. Le contour n'apparaît pas clairement. C'est en fait pour ne pas surcharger le dessin. Une fois que le contour est bien défini, l'utilisateur doit appuyer sur le bouton "OK". Lorsque l'on sortira de la boîte de dialogue de réglage d'une icône, le contour sera créé. Il ne sera bien sûr pas visible, mais un click de la souris hors de ce contour ne provoquera pas de réaction de la part du programme.

4.3.1.d. Le réglage des moyens d'interaction

Jusqu'en septembre 1993, les moyens d'interaction étaient limités au clavier et à la souris. Toutefois, notre prédécesseur avait pris soin de prévoir d'autres moyens d'accès au programme. En effet, dans la boîte de dialogue préexistante, il existait une possibilité d'interagir avec un joystick ou un switch (contact unique). Malheureusement, au-delà de cette boîte, il n'y avait rien. On ne pouvait faire aucun réglage. Or dans un logiciel pour personnes handicapées, il est primordial de pouvoir paramétrer aussi bien la vitesse de la souris que la vitesse de défilement des icônes quand on utilise un switch. C'est à quoi nous nous sommes attachés : rendre accessible le programme au moyen du switch (en priorité) car c'est le moyen d'interaction le plus utilisé au centre "La Famille", et pouvoir régler tous les moyens d'interactions.

Prenons ces moyens d'interaction dans l'ordre où ils apparaissent. Tout d'abord le clavier. Nous nous proposons de régler la vitesse de répétition d'une touche. Cela pourrait se faire par la boîte de dialogue ci-dessous :

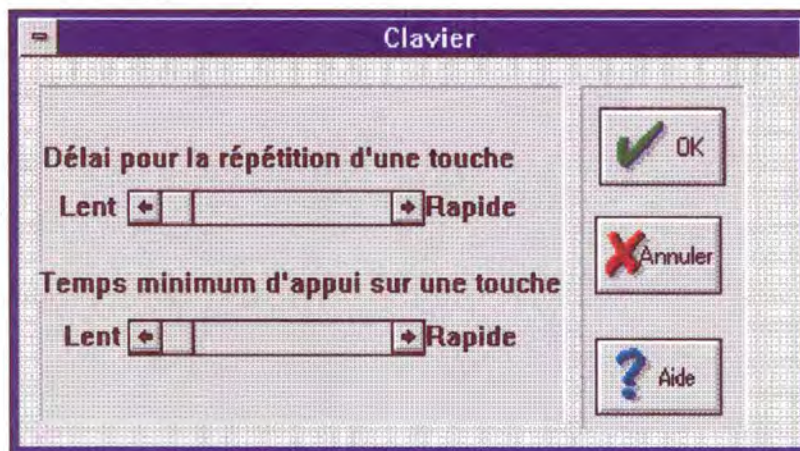


Figure 4.4. : Réglage du clavier.

Le délai pour la répétition d'une touche est le temps qui sépare deux envois du même caractère vers le buffer du clavier, lorsqu'on laisse le doigt posé sur la touche. Le temps minimum d'appui sur une touche permet de faire un réglage pour les personnes dont les gestes sont saccadés, de manière à ne pas faire de fausse manoeuvre.

Pour ce qui touche la souris, il était prévu que l'on puisse régler la vitesse du curseur et la vitesse du double click. Nous avons estimé, puisque c'est maintenant possible, qu'il serait agréable de pouvoir régler séparément les vitesses horizontales et verticales. Nous avons donc défini une nouvelle boîte de dialogue :

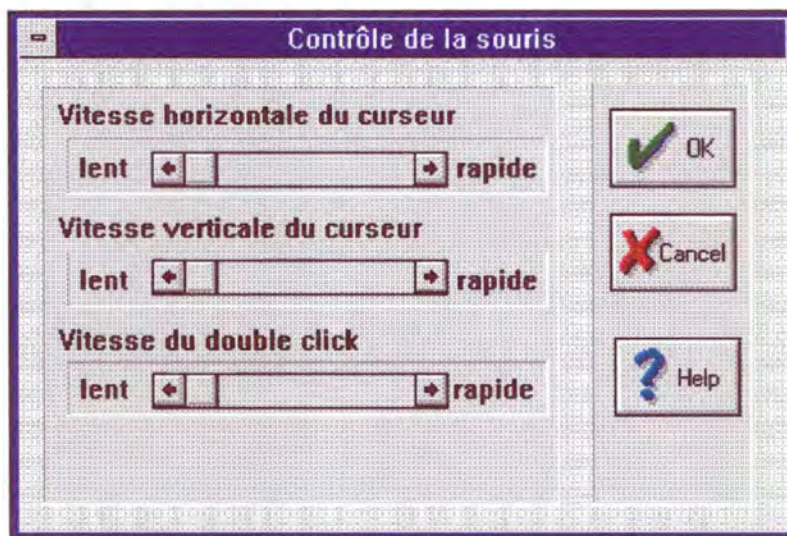


Figure 4.5. : Gestion des paramètres de la souris.

Le réglage séparé des vitesses horizontales et verticales est un concept nouveau que Microsoft a apporté dans son dernier pilote de souris. C'est un concept assez intéressant pour une personne qui ne maîtrise l'espace que dans une seule direction. Cela lui permet de se déplacer dans la direction qu'elle maîtrise avec beaucoup de facilité et dans la direction qu'elle ne maîtrise pas, avec une égale facilité puisque le curseur va plus lentement.

Pour le joystick, on peut régler plusieurs paramètres. Cela va de la forme du curseur au calibrage du joystick en passant par le réglage de la vitesse du curseur. La boîte de dialogue suivante règle le problème :

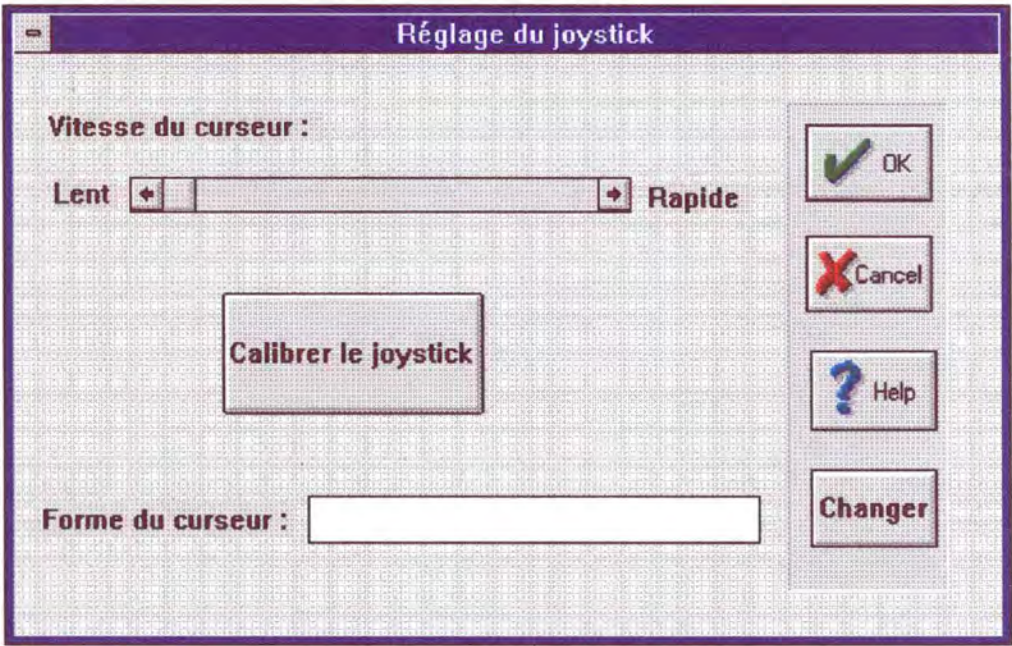


Figure 4.6. : Gestion des paramètres du joystick.

Enfin, pour le contact unique (switch), Luc Vandenaabeele signale plusieurs éléments à prendre en compte dans le réglage : le délai de défilement, le délai de répétition entre deux actions et le temps minimum d'appui pour une action⁶. La boîte de dialogue qu'il propose dans son mémoire convient parfaitement à nos besoins :

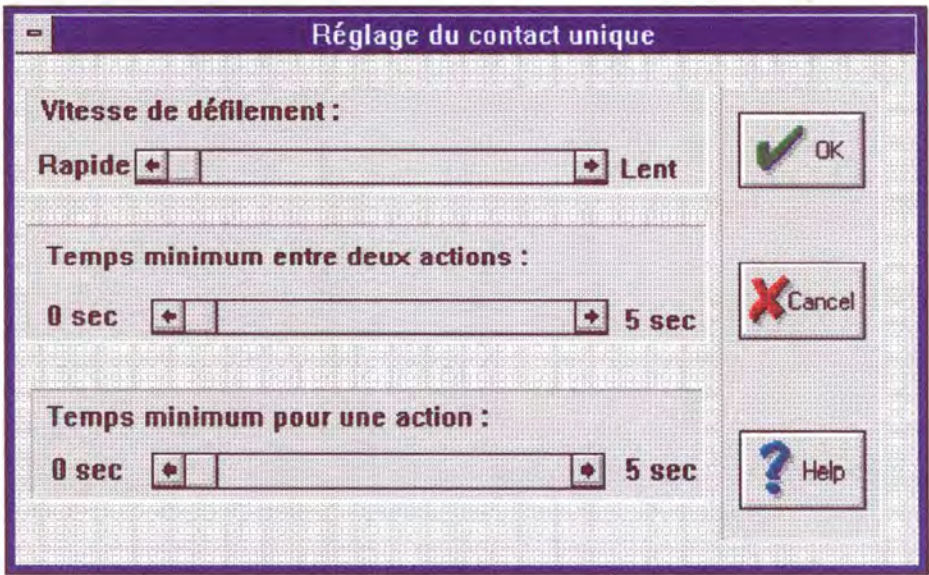


Figure 4.7. : Gestion des paramètres du switch.

⁶ Cf. [VAND,93],p.116

4.3.1.e. Le commentaire du code de monsieur Luc Vandabeele

Le commentaire du code de Luc Vandabeele a été un de nos premiers soucis. Avec son aide, nous avons commenté la plupart des méthodes écrites. C'était primordial, d'abord pour la bonne compréhension du programme, et ensuite pour la réutilisation des méthodes dans d'autres situations. Seules quelques méthodes n'ont pas été commentées : ce sont toutes celles dont nous n'avons pas eu besoin.

4.3.2. Les solutions proposées pour les besoins secondaires

4.3.2.a. l'analyse de la trace

L'analyse de la trace existant pour le moment est très sommaire. Elle consiste à placer dans un fichier texte les différents clicks de l'élève, qu'ils soient pertinents ou pas. C'est au responsable de l'exercice de calculer les taux de réussite et d'échec de l'élève.

En concertation avec le centre "La Famille", nous avons défini des pistes de réflexion pour que la trace soit parlante pour les éducateurs. Ainsi, pour le clavier, la souris et le contact unique, certains chiffres sont plus importants que d'autres : ce sont, par exemple, le nombre total de clicks ou de sélections, le pourcentage de clicks valables, le pourcentage de clicks valables mais non pertinents, le pourcentage de clicks valables et pertinents, le temps total mis pour compléter l'exercice,... De plus, il faut représenter ces valeurs de manière visuelle. Et de ce côté, peu importe le moyen, il n'y a qu'un objectif : il faut que cette représentation soit PARLANTE.

Pour le joystick, l'analyse de la trace est un peu différente : d'après les remarques faites au centre "La Famille", il serait pratique de pouvoir visualiser où le curseur du joystick s'est baladé pendant l'exercice. Pour cela, l'analyse de trace doit se faire graphiquement. La solution que nous proposons est de trier les messages en provenance du joystick (notamment tous les messages de déplacement) et de les stocker pour une utilisation ultérieure. En théorie, il suffirait de stocker ces messages, puis de les relire en les traçant dans une fenêtre. On pourrait alors analyser les problèmes de manipulations,...

En discutant le cahier des charges, il nous est apparu que cette méthode pouvait aussi convenir pour la souris. Toutefois, la souris étant nettement plus mobile qu'un joystick, nous émettons de sérieuses réserves quant à ce procédé avec la souris.

4.3.2.b. Le mode d'emploi

Le mode d'emploi est primordial dans un projet comme celui-ci. Nous en fournissons un exemplaire dans les annexes de ce mémoire. Il est primordial afin que les gens qui devront se servir d' "Auteur !" puissent utiliser plus de 10% du logiciel. Nous préparons un mode d'emploi reprenant les différents concepts présentés dans le logiciel ainsi que leur utilisation. Ce mode d'emploi ne remplacera cependant jamais une bonne aide en ligne dans le programme. Malheureusement, celle-ci, pour être bien faite (c'est-à-dire efficace et valable) demande un temps dont nous ne disposons pas.

Ce mode d'emploi devrait aussi contenir les instructions permettant d'ajouter des sons et des images dans la bibliothèque existante ainsi que les bases d'une méthodologie de création d'exercices (chose demandée par le centre "La Famille" afin d'encourager les plus réticents à utiliser le programme).

4.3.2.c. La compatibilité d'écrans

Les configurations suivantes ont été testées avec succès :

- ♦ EGA,
- ♦ VGA (640x480),
- ♦ SVGA (800x600),
- ♦ SVGA (1024x768).

Le nombre de couleurs n'a de l'influence que sur la netteté du dessin et donc n'entre pas en ligne de compte dans le programme.

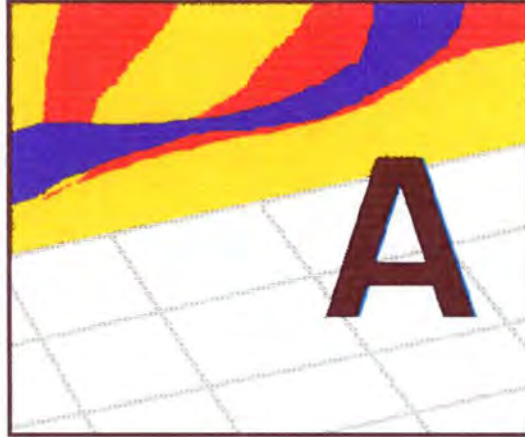
4.3.2.d. La navigation d'objets

Un des buts du centre est d'utiliser "Auteur!" afin de créer des circuits que les utilisateurs devraient parcourir. Ces circuits seraient en fait une suite d'écrans comportant des lignes figurant les limites du circuit. De ce but découle un petit problème : la possibilité de tracer des lignes n'existait pas dans la première version d' "Auteur!". Nous avons donc ajouté ce concept au programme. Mais nous nous sommes vite rendus compte que la ligne en elle-même ne suffisait pas.

Il nous fallait aborder le problème du circuit et de la ligne séparément. La ligne et le circuit seront des dérivés de l'objet "élément". La ligne sera paramétrable mais non sélectionnable (sélectionner une ligne relève dans la plupart des cas très difficile). Donc il ne pourra y avoir aucune action associée à une ligne. Elle sera là comme séparateur. Le circuit sera constitué de trois régions : la région de rejet, la région de dégagement, la région du circuit. Nous reviendrons plus tard sur ce nouveau concept.

4.4. Conclusion de ce chapitre...

Comme nous l'avons vu, il y avait beaucoup de problèmes à traiter. Beaucoup subsistent encore, mais les plus importants sont terminés. Le fait de travailler pour une population spéciale implique l'apparition de nouveaux problèmes qu'il est important de traiter si l'on veut que notre travail serve à quelque chose. Il y a en effet beaucoup de programmes qui existent sur le marché informatique, mais la plupart d'entre eux sont en général assez fermés et dirigés vers une tranche bien spécifique de la population. Notre but était de faire un programme assez ouvert et relativement bien compréhensible et accepté de tous. Nous pensons avoir atteint en cela notre but : le programme s'adresse à une large gamme d'utilisateurs, et sa simplicité le rend accessible à tous.



Chapitre 5

Outils et difficultés de programmation

Où l'on décrit l'environnement utilisé et les problèmes provoqués par cet environnement et où l'on s'aperçoit que la programmation Windows n'est pas aussi simple qu'elle n'y paraît.

5.1. Windows

Windows, comme son nom l'indique, est un système d'exploitation basé sur un système multi-fenêtré. Depuis 1985, année de sortie de la première version, bien des choses ont changé dans cet environnement. A l'heure actuelle, la technologie et les architectures 32 bits profitent pleinement du "moteur" de Windows. Dans quelques mois sortira une nouvelle version encore plus révolutionnaire : Windows 4.0 "Chicago". Cette nouvelle mouture servira de système d'exploitation à part entière. Les habitués du DOS et de la ligne de commande ne verront donc plus leur environnement préféré qu'au travers d'une fenêtre de Windows.

Cet environnement, que les utilisateurs du MacIntosh connaissent bien, déjà depuis de nombreuses années, est de plus en plus populaire auprès des utilisateurs de PC's. Nous en voulons pour preuve le nombre de nouvelles applications qui sortent sous cet environnement. Même les éditeurs de jeux vidéo profitent du multi-fenêtrage offert pour ressortir un vieux programme avec la mention "SOUS WINDOWS" en grand et en travers sur la boîte. Autre exemple, et il est encore plus frappant, il n'arrive plus aucun traitement de texte sur les marchés informatiques qui ne soit pas sous Windows : la richesse des caractères, la possibilité d'intégrer au sein d'un même document des graphiques, des images, des équations et même du son (depuis la version 3.1) font des traitements de texte sous MS-DOS des parents pauvres de ce genre de programme.

Bref, Windows (et son environnement fenêtré) est à la mode. Mais ce n'est pas qu'une question de mode. Avoir tout, traitement de texte, tableur, logiciel graphique,... sous la main est tentant pour n'importe quel utilisateur. Il est plus facile de cliquer deux fois sur une icône représentant le programme plutôt que de se souvenir de son nom et de son chemin d'accès. Tout le monde sait faire un double click : même des gens qui ne comprennent pas grand chose à l'informatique, peuvent le faire. Le succès d'un tel système est dès lors bien compréhensible. La facilité de lui attacher de nouveaux périphériques (imprimante, scanner, vidéo, carte sonore,...) y est aussi pour beaucoup, quoique elle n'atteint pas encore le niveau du "*plug and play*" existant sur le MacIntosh. Toutefois, la nouvelle version devrait encore améliorer ce concept du "*plug and play*".

Cette facilité et ce succès ne nous ont pas laissés indifférents. C'est une des raisons pour lesquelles nous avons décidé de porter le programme "Auteur !" sous l'environnement Windows. Il faut avouer aussi que nous avons choisi Windows pour des raisons de facilité d'accès au langage de programmation (ils sont en effet légion, les éditeurs de compilateurs qui offrent des compilateurs sous Windows) mais nous y reviendrons dans la section suivante consacrée à la programmation Windows, car la programmation Windows, si elle est facile d'accès, est loin d'être facilement abordable sans une solide formation.

5.2. La programmation Windows

La programmation Windows n'est pas une chose facile à assimiler. Il faut, si on a les notions de programmation orientée objet, entre 3 et 4 mois pour commencer à bien comprendre tous les concepts et les mécanismes de Windows.

L'objet de cette section n'est pas de donner un cours sur la programmation Windows. Toutefois, il nous faut parler de certains concepts afin de bien comprendre à quoi nous avons été confrontés.

Premièrement, on ne peut vraiment, efficacement, programmer sous Windows sans passer par la programmation orientée objet (P.O.O.). Cela nécessite de se familiariser avec cette philosophie de programmation relativement récente. Nous n'expliquerons pas en quoi elle consiste, il faudrait pour cela un mémoire à part entière. Simplement, il faut être familier avec les techniques d'héritage, d'instanciation,... Bref, les notions de base de la P.O.O.

Pour illustrer ces notions de base, prenons un petit exemple. Imaginons que nous disposions d'un objet "cylindre", ayant comme caractéristiques une hauteur, un rayon, un volume,... Supposons également que l'on puisse effectuer les actions suivantes sur cet objet : on peut le remplir, le vider, le casser,...

Une instance de cet objet cylindre pourrait être l'objet bouteille : en effet, une bouteille est caractérisée par une hauteur, un rayon, un volume (exprimé par sa contenance),... Et une bouteille peut être cassée, remplie, vidée,...

De même, l'objet bouteille pourrait être un dérivé de l'objet cylindre : il hériterait de toutes les caractéristiques de l'objet cylindre mais en plus, il pourrait avoir d'autres données et d'autres méthodes (actions que l'on peut entreprendre sur l'objet). Par exemple, la bouteille pourrait contenir une donnée qui dirait si elle est consignée ou pas, si elle est recyclable ou pas,... Des actions supplémentaires pourraient être définies : on peut servir une bouteille, renverser une bouteille,...

Deuxièmement, il faut savoir que Windows traite tous les programmes grâce à des messages. Cela peut se représenter comme suit :

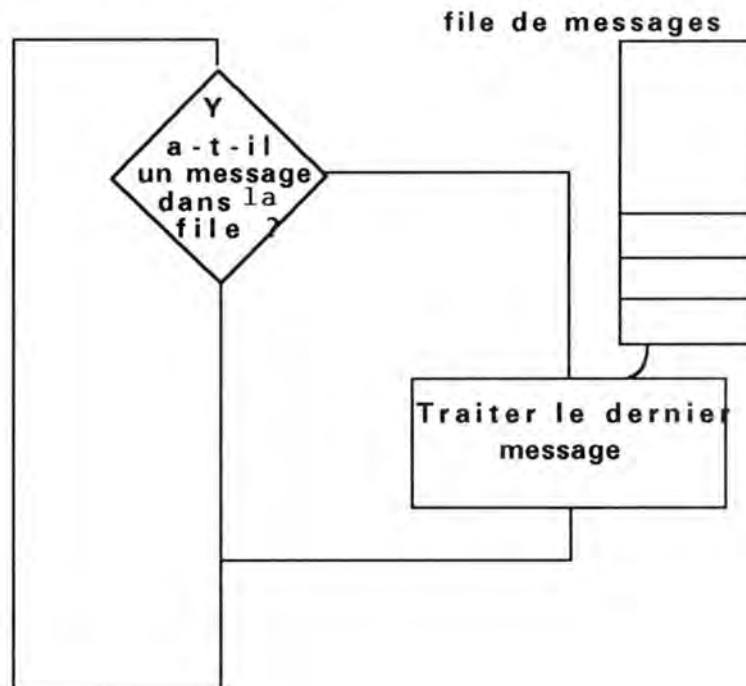


Figure 5.1. : La traitement des messages par le moteur de Windows.

Chaque programme envoie des messages dans une file d'attente. Une fois dans la file, ces messages sont pris en compte normalement dans l'ordre où ils sont arrivés (il peut arriver, mais c'est assez rare, qu'un programme demande le traitement prioritaire d'un message). Ainsi, un double click sur une icône est vu par Windows comme une suite de 4 messages :

- le bouton de la souris est enfoncé (1° message).
- le bouton de la souris est relâché (2° message).
- un chronomètre est mis en route (3° message).
- le bouton de la souris est enfoncé (4° message).

Windows n'a donc plus qu'à comparer le temps inscrit au chronomètre avec le temps figurant dans les variables d'environnement, temps qui correspond à la vitesse de double click. Si le temps chronométré est plus petit, le double click est pris en compte, s'il est plus grand, c'est comme s'il y avait eu deux clicks séparé sur l'icône. Ce petit exemple montre bien le phénomène des messages sous Windows.

Troisièmement, il faut savoir comment Windows gère son écran. Il faut savoir comment Windows fait pour savoir où il a mis les choses qu'il range.

Pour Windows, une fenêtre est un objet, et à un objet correspond un pointeur. C'est donc une chaîne de pointeurs qu'il faut parcourir à tout moment lorsqu'on veut atteindre un élément bien précis de l'écran (ce qui est fréquemment le cas dans "Auteur !"). Heureusement, cette chaîne de pointeurs est gérée entièrement par Windows. En expliquant cela, nous expliquons la façon de sauver un écran sous Windows, pour autant que les éléments de l'écran soient des fils ou des filles de l'écran. Voyons un peu la structure d'un écran : Imaginons un écran d' "Auteur !" composé comme suit :

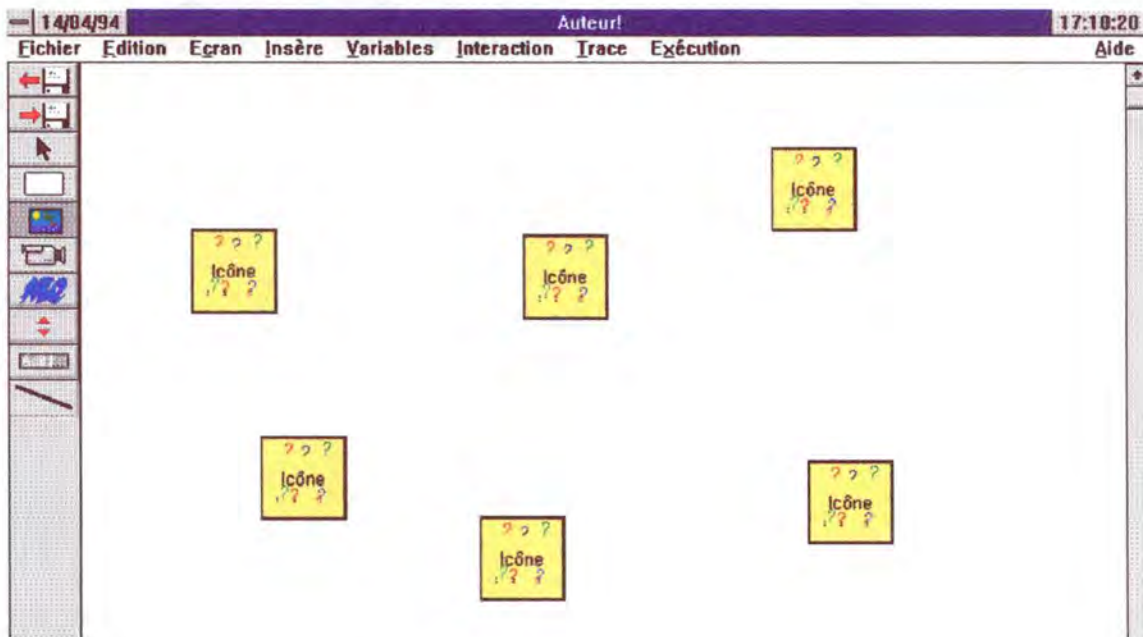


Figure 5.2. : A chaque fenêtre correspond un pointeur

La chaîne de pointeur peut se représenter comme suit :

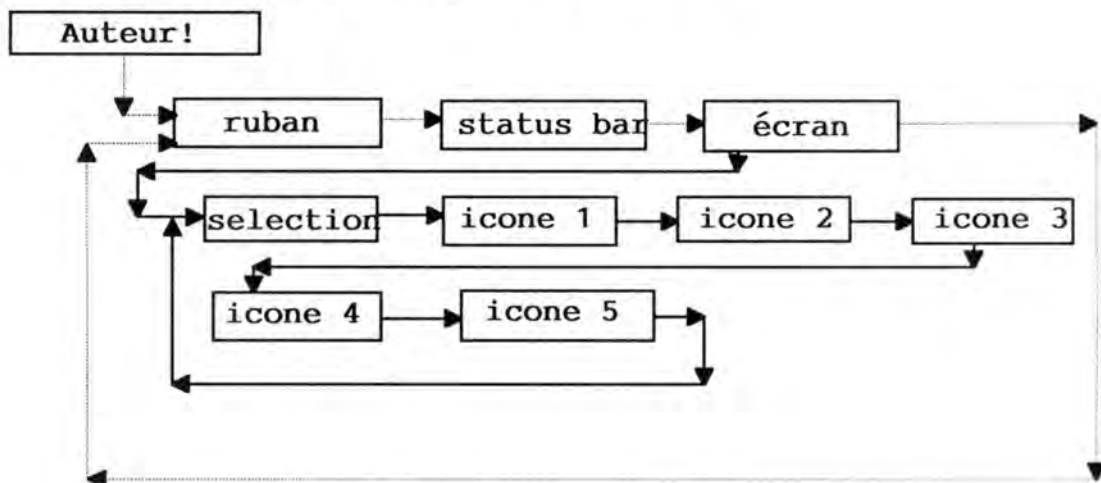


Figure 5.3. : La chaîne de pointeur des éléments de l'écran.

Cette chaîne est donc une liste doublement chaînée circulaire. C'est ce qu'on fait de mieux en matière de liste. De plus cette gestion des fenêtres est complètement transparente à l'utilisateur. Elle ne peut malheureusement pas le rester au programmeur qui, lui, doit savoir comment les fenêtres sont gérées.

Ce concept n'est, malheureusement, pas très bien expliqué dans les divers documents de programmation. On peut juste trouver des allusions à ce concept de "pointeur vers un objet fenêtre" dans l'aide du Borland C++ avec lequel nous avons travaillé.

5.3. Les difficultés rencontrées...

Dans cette section, nous allons présenter les diverses choses que nous avons entreprises sur le logiciel "Auteur!" et les difficultés s'y rapportant. Ces choses sont dans l'ordre : la sauvegarde et le chargement, le déplacement d'objet, les moyens d'interaction, la ligne et le texte, les circuits, et enfin le joystick.

5.3.1. La sauvegarde et le chargement

La sauvegarde part du principe de liste que nous avons expliqué ci-dessus. Nous n'allons pas nous étendre inutilement sur ce concept. Nous signalons simplement la façon dont nous avons procédé. C'est donc en quelque sorte un algorithme que nous présentons ici.

En fait, pour la sauvegarde, nous avons prévu, dans chaque objet, une méthode de sauvegarde et une méthode de chargement. La méthode de sauvegarde sauve une structure de données remplie au moment de la création de l'objet et mise à jour chaque fois que l'objet changeait. La méthode de chargement faisait la même chose mais dans l'autre sens : elle garnissait les champs d'une structure de données servant à créer l'objet. C'est ici que nous nous sommes heurtés à une difficulté, tout à fait logique quand on y réfléchit.

La sauvegarde des données de l'objet ne posait aucun problème : l'objet était créé, les données étaient dans la structure,... Bref, la seule chose à laquelle il fallait faire attention, c'est que les bonnes données figurent dans la structure. Là où le problème se posait, c'était pour le chargement. En effet, comment appliquer une méthode à un objet qui n'existe pas. C'est impossible sauf en créant l'objet vide puis en lui changeant ses paramètres. Mais cette façon de faire était très lourde. Nous avons donc dû oublier les méthodes de chargement, au profit d'une méthode générale, de l'objet application, ne servant qu'au chargement et à la création des objets sauvegardés sur disque. Cette méthode, si elle est à peine plus légère que l'autre, a l'avantage de rassembler tout le chargement en un seul endroit. Il est donc plus facile, en cas d'erreur, de voir où il y a un problème.

La méthode de sauvetage générale se dessine comme suit :

```
Créer un fichier pour la sauvegarde.
Sauvegarder tous les paramètres d'environnement (moyen d'interaction sélectionné,
réglages,...)
Compter le nombre d'écrans
Ecrire ce nombre d'écrans
Pour ce nombre d'écrans, faire la suite d'instructions suivante :
    Prendre le premier écran
    Ecrire le type d'objet
    Ecrire le nombre d'éléments contenus dans cet écran
    Appeler la méthode de sauvegarde d'un écran
    Pour le nombre d'éléments faire la suite d'instructions suivante :
        Appeler la méthode de sauvegarde de l'élément
        Si l'élément possède des actions
            Appeler la méthode de sauvegarde de la liste
            d'actions.
    Passer à l'élément suivant
Passer à l'écran suivant
Fermer le fichier.
```

Figure 5.4. : algorithme de sauvegarde d'un exercice.

La méthode de chargement est sensiblement la même, mais à l'envers. La seule différence est que, plutôt que d'appeler des méthodes de chargement, on exécute un chargement dans des variables temporaires qui, par la suite, servent à créer les objets.

Les constructeurs des objets ont donc dû être modifiés en conséquence, de manière à pouvoir accepter plus de paramètres.

Les méthodes de sauvetage et de chargement d'un seul écran sont les mêmes, excepté qu'on ne sauve pas le nombre d'écrans, et les paramètres d'environnement. Il y donc une boucle en moins. De plus, la méthode de chargement fait appel à une méthode de "nettoyage" de l'écran. Ce "nettoyage" ne se fait pas quand on charge un seul écran : celui-ci vient s'ajouter à la suite d'écrans qui existent déjà.

5.3.2. le déplacement d'objet

Le déplacement d'objet n'a pas créé de problèmes particuliers, si ce n'est un problème de message que l'on retrouvait également lorsqu'on chargeait un exercice du disque. Le problème était en fait que les exécutions d'actions ne se faisaient plus. L'explication du phénomène est en fait assez simple. L'exécution d'une action se fait en fait par l'envoi d'un seul message. Ce message n'est pas un message spécifique de Windows. Il s'agit d'un "WM_USER message" qui est défini par l'utilisateur, c'est-à-dire nous dans ce cas précis. A cet envoi de message correspond une méthode qui intercepte le message en question. Hors nous avons vu que les messages sont stockés dans une file d'attente. Ce que nous n'avions pas compris tout de suite, c'est que chaque objet a sa file d'attente pour les messages qui lui sont propres. Dans le cas du chargement et du déplacement, le message était envoyé à un objet qui n'existait plus. Et donc, la méthode interceptant ce message n'était jamais appelée puisque le bon objet, celui qui existait bel et bien, ne recevait pas le message en question.

Nous avons résolu ce problème en envoyant le message à toutes les files d'attente. Nous émettons ce message de telle sorte que toutes les files d'attente le reçoivent, et seules les méthodes d'interception concernées prennent ce message en compte. C'est une solution assez "laide", mais elle a l'avantage de fonctionner parfaitement et de ne pas nous demander trop de temps. Une méthode "plus propre", serait de sauver le numéro de la file d'attente où il faut envoyer le message. Ainsi, cela éviterait d'avoir des files d'attentes supplémentaires qui ne servent plus à rien.

5.3.3. Les moyens d'interaction

Nous n'avons pas eu de gros problèmes pour mettre en oeuvre les différents moyens d'interaction dont nous avons déjà parlé dans un chapitre antérieur. Les seules difficultés que nous avons rencontrées sont d'ordre documentaires. En effet, modifier la vitesse du curseur de la souris ne se fait pas par une quelconque API (Application Programming Interface : sorte de macro-instruction très souvent utilisée dans Windows dont elles sont la base) nommée "ChangeMouseSpeed".

A la place, il y a bien une API responsable de tous les paramètres de Windows : vitesse du curseur de la souris, vitesse de double-click, vitesse de répétition d'une touche du clavier,... Cette méthode s'appelle SystemParametersInfo. Et selon l'argument qu'on lui passe, elle renvoie la valeur d'un paramètre système ou elle change une valeur des paramètres du système. C'est cette méthode que nous avons utilisée. C'est, à notre connaissance, la seule méthode possible pour faire ce genre de chose.

La documentation de cette API est encore une fois bien cachée et il faut vraiment chercher pour y avoir accès.

La deuxième difficulté était de trouver un pilote pour un joystick sous Windows. Nous en avons finalement trouvé un, produit par IBM. Le joystick ne répond pas, en fait, au message de la souris, comme nous le pensions au départ. Nous reviendrons sur la gestion du joystick ultérieurement.

5.3.4. La ligne et le texte

Pour la ligne, c'est à un problème d'affichage que nous nous sommes heurtés. En effet, la ligne étant un dérivé de l'objet "élément", elle devait obligatoirement s'inscrire dans une fenêtre. Or au départ, nous ne voulions pas voir cette fenêtre. Nous avons donc mis sa taille à 0. Et donc, la ligne s'inscrivant dans la fenêtre, elle ne s'affichait pas. Il nous a donc fallu, donner une taille à la fenêtre pour pouvoir afficher une ligne. Nous avons eu le même problème pour afficher du texte.

Autre problème, quand on veut afficher dans une fenêtre il faut préciser les coordonnées auxquelles on veut afficher le texte ou la ligne. Or, si on veut que le texte soit apparent, il faut mettre ces coordonnées d'affichage à zéro. Sinon, l'affichage se fait bel et bien mais en dehors de ce qui est visible. Et donc cela donne l'impression que l'affichage ne se fait pas. C'est un problème qui nous a causé bien du souci pour peu de chose ; notre méthode était bonne, mais le texte ou la ligne n'était pas affiché au bon endroit et donc n'apparaissait pas.

5.3.5. Le détournage

Nous avons déjà eu l'occasion de citer les problèmes que nous avons eus avec le détournage des pictogrammes servant à garnir les icônes. Nous ne reviendrons donc pas sur le sujet. Un détail toutefois, pour trouver comment afficher un fichier BITMAP dans une boîte de dialogue, il faut se transformer en Sherlock Holmes des réseaux informatiques et épilucher des tonnes d'articles et de questions posées par des centaines de gens. Bref, il semble que c'est une question tabou. C'est pourtant un concept qui devient relativement courant.

5.3.6. Les circuits

Les circuits nous ont posé d'autres problèmes. Avant tout des problèmes de conception. Le challenge était de pouvoir faire naviguer un curseur dans un espace clos, en lui assignant des actions selon la zone où il se trouvait.

Nous avons trouvé une solution grâce aux régions gérées par Windows. En effet, on peut, à partir d'un nuage de points, définir une région (c'est ce concept que nous avons employé pour le détournage des objets). Mieux encore, on peut tester si le curseur se trouve ou non dans cette région. Il suffit alors, à partir de ces éléments, de définir un circuit comme un assemblage de régions. Chaque région ayant sa ou ses propres actions. Une région n'apparaît pas à l'écran tant que l'on ne lui en donne pas l'ordre. Il existe des régions de base (rectangulaires, elliptiques et polygonales) et des régions de formes complexes, qui sont la combinaison des autres. On peut unir deux régions, les exclure mutuellement l'une de l'autre, faire un "et" ou un "ou" logique de deux régions,...

5.3.7. Le joystick

Le joystick, nous l'avons déjà signalé, est un accessoire particulier. Les gens de chez Microsoft n'ont visiblement pas prévu que l'on pourrait s'en servir dans Windows (en tout cas pas dans cette version). Le joystick est rangé dans les appareils multi-médias et donc est géré en tant que tel. Il faut donc d'abord charger le *driver* du joystick en mémoire (surtout ne pas oublier de le décharger à la fin du programme) puis intercepter tous les messages en provenance du joystick. Mais ce n'est pas tout, il faut pouvoir interpréter les messages. Quand on fait bouger le manche du joystick, il faut savoir dans quel cadran on se trouve. En effet, le joystick se décompose en quatre cadrans. Chaque cadran possède une valeur minimale et une valeur maximale. C'est en fonction des valeurs renvoyées en paramètres du message que l'on sait dans quel cadran on se trouve et donc dans quelle direction on doit bouger.

L'intensité avec laquelle il faut bouger le curseur est normalement fournie par un autre paramètre du message. Nous avons décidé, dans un premier temps (afin de d'abord bien comprendre comment ça fonctionne) de ne pas jouer sur l'intensité du mouvement

A remarquer que sur une autre machine, nous n'aurions pas besoin de jouer sur l'intensité du mouvement. Ceci est dû au fait que les joysticks PC sont des joysticks digitaux et non pas analogiques comme sur la plupart des autres machines.

La figure suivante montre ces quatre cadrans avec leurs valeurs minimales et maximales respectives :

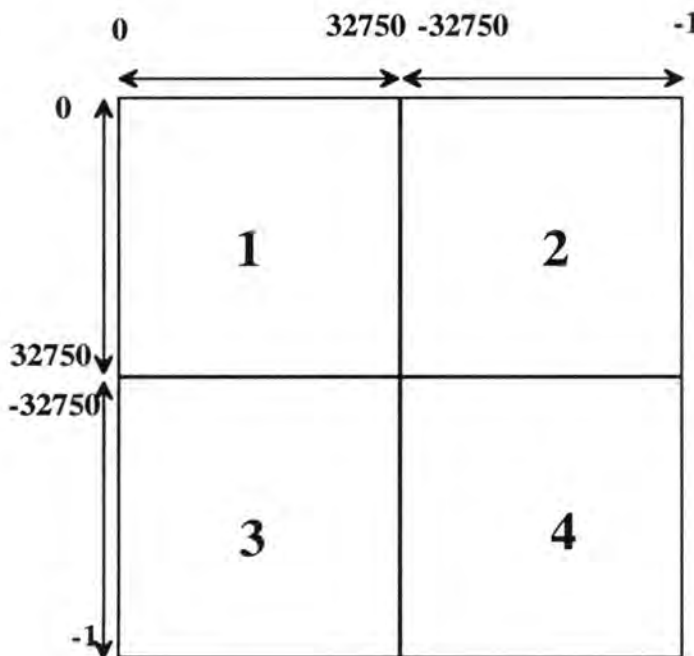


Figure 5.5. : Les quatre cadrans du joystick

On remarque donc bien, dans cette figure, les valeurs minimales et maximales de chaque cadran. Ces valeurs sont représentées par des entiers. Ce qui explique les valeurs que l'on trouve en regardant le graphique. Il est important de remarquer que les valeurs maximales sont parfois un rien différentes. Nous pensons que, normalement, ces valeurs maximales sont égales à la limite supérieure des entiers (à savoir : -32768 si c'est négatif et 32767 si c'est positif). Le fait que nous n'obtenions pas ces valeurs est, à notre avis, dépendant du moment où le message donnant la position est envoyé.

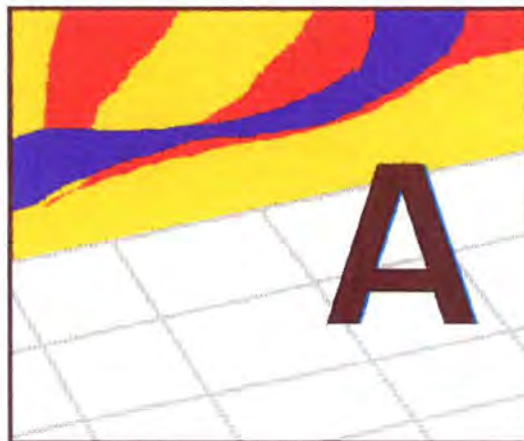
A titre d'information, nous donnons les coordonnées minimales et maximales de chaque cadran. Le premier cadran est compris entre les coordonnées $\langle 0,0 \rangle$ (en haut à gauche) et $\langle 32750,32750 \rangle$ (en bas à droite). Le deuxième cadran est compris entre les coordonnées $\langle -1,0 \rangle$ (en haut à droite) et $\langle 32750,-32750 \rangle$ (en bas à gauche). Le troisième cadran est compris entre $\langle -1,0 \rangle$ (en bas à gauche) et $\langle -32750,32750 \rangle$ (en haut à droite). Le quatrième, enfin, est compris entre $\langle -1,-1 \rangle$ (en bas à droite) et $\langle -32750,-32750 \rangle$ (en haut à gauche).

Les deux instructions pour charger/décharger les *driver* en mémoire (que ce soit celui du joystick ou un autre) sont, qui l'eût cru, *OPENDRIVER* et *CLOSEDRIVER*.

5.4. En conclusion de ce chapitre

En conclusion de ce chapitre consacré aux difficultés de programmation de Windows, nous dirions que programmer sous Windows demande du temps. Nous avons, grâce à ce projet, touché un peu tous les domaines de la maintenance : les ajouts, les corrections, les changements,... Nous avons également, mais ce n'était pas un concept entièrement nouveau pour nous, touché à la programmation orientée objet. La grande nouveauté a été sans conteste l'apprentissage d'un nouveau langage, le C++. C'est, en effet le seul langage avec lequel il est "facile" de programmer sous Windows (n'oublions pas que Windows est, en grande partie, écrit en C).

Pour le reste, les difficultés auxquelles nous avons été confrontés prouvent bien que la programmation sous Windows n'est pas une chose aisée. Toutefois, avec un peu d'habitude, de pratique et beaucoup, beaucoup d'exercices, c'est un mode de programmation à la portée de tous les informaticiens et de tous ceux qui s'intéressent d'un peu plus près à la programmation d'applications.



Chapitre 6

Une méthode de construction d'exercices

Où l'on découvre certaines possibilités offertes par "Auteur !" au moyen d'un exemple simple du jeu "Chercher l'intrus". On découvre une ébauche de méthode permettant de construire de nombreux exercices.

6.1. Introduction

Lors des rencontres avec le centre "La Famille", il nous a été demandé de pouvoir donner une méthode de construction d'exercices pour le logiciel "Auteur!". Cette méthode devait aider les éducateurs et thérapeutes encore effrayés par le diable informatique et son suppôt, l'ordinateur. Mais les aider pour quoi ? Il est vrai que pour construire des exercices avec le programme "Auteur !", il faut souvent raisonner comme un informaticien quand il construit un algorithme. Il y a des choix à faire, des boucles à implémenter,... Bref, toutes sortes de notions purement informatiques.

Nous sommes conscients que ces concepts ne peuvent être abordés sans guidance. Cette guidance, nous nous proposons de la faire par l'intermédiaire de cette "méthodologie". Nous allons montrer comment faire des exercices simples, comment combiner des exercices simples pour faire des exercices plus complexes,... Notre démarche sera la suivante : nous partirons de la conception papier d'un exercice mono-écran, puis nous compliquerons avec l'adjonction d'autres écrans et des enchaînements s'y rapportant. Nous passerons les différentes étapes sur l'ordinateur en expliquant les gestes à faire pour arriver à un résultat plus ou moins satisfaisant.

6.2. A vos crayons...

6.2.1. Un effort d'imagination...

La première étape est un effort d'imagination. En effet, ce n'est pas parce qu'on a un ordinateur que celui-ci va faire tout à la place de l'utilisateur. Celui-ci peut lancer le programme pendant 237 jours d'affilée, il n'en sortira rien de plus que ce qu'il y aura mis. Il faut donc, premièrement, concevoir les exercices dans la tête et puis seulement s'asseoir devant... une feuille de papier... Il faut d'abord imaginer l'exercice comme si l'on ne disposait pas d'un ordinateur. Ensuite, on voit si, vu que l'on a un ordinateur, ce genre d'exercice est possible avec "Auteur!". Il faut faire un plan.

6.2.2. Un plan sur papier...

Le plan sur papier est la deuxième étape. Il va falloir bien décrire sur une feuille toutes les étapes, tout ce qui sera des écrans, des icônes, des actions,... Il faut également écrire les enchaînements entre les éléments. Nous proposons de suivre la démarche suivante : lorsque l'exercice est clair dans la tête, il faut en faire une maquette sur papier. Les petits défauts auxquels on ne pense jamais apparaissent alors.

Pour reprendre un exemple bien connu, nous allons développer un exercice, déjà présenté lors du premier chapitre : c'est l'exercice de l'intrus. C'est un exercice simple, que tout le monde connaît. L'effort d'imagination n'est donc pas trop grand. Pour bien fixer les idées, l'exercice de l'intrus que nous avons en tête comporte une série de quatre images. Parmi celle-ci se trouve une image qui n'a aucun rapport avec les autres. C'est cette image qu'il faudra que l'enfant sélectionne pour réussir l'exercice. Toute autre sélection entraînera un échec et un retour au début de l'exercice.

Maintenant que l'exercice est clair dans la tête de chacun, nous allons poser ses bases sur papier. Il nous faut donc répondre à certaines questions : quels pictogrammes va-t-on employer ? Qu'est-ce que l'enfant connaît ? Connaît-il les couleurs ? les formes ? Où va-t-on placer les pictogrammes sur l'écran ? en haut, au milieu, en bas, ... ? De quelle couleur doit être le fond de l'écran ? ...

Il faut donc régler tous les petits détails. Dans un premier temps, nous n'utiliserons qu'un seul écran. Une nouvelle question apparaît : si on n'utilise qu'un seul écran, comment va-t-on faire les renforcements positifs et négatifs ? Voyons cela en détail.

Nous voulons faire un exercice mono-écran reprenant le principe du jeu de l'intrus. Nous allons placer les dessins au milieu de l'écran en une ligne. Le fond de l'écran devrait avoir une couleur neutre (blanc ou blanc cassé). Nous allons établir un jeu de l'intrus où l'intrus sera une personne de sexe différent. Il y aura trois dessins de femme et un dessin d'homme. Nous supposons que l'enfant ne connaît pas les couleurs : nos dessins seront tous bleutés. Nous supposons que l'enfant est capable de différencier un homme d'une femme, ne serait-ce que par l'aspect vestimentaire.

Pour choisir l'intrus, l'enfant devra cliquer, à l'aide de la souris, sur l'image qu'il croit ne pas appartenir à la série. Bien sûr, il existe, dans "Auteur !", d'autres moyens d'interactions. On peut définir que l'enfant devra utiliser le clavier pour sélectionner la bonne image, ou le joystick, ou un balayage automatique de l'écran.

Pour les renforcements, nous prévoyons un son associé à chaque pictogramme. Ces sons seront respectivement : un cri de joie pour la réussite, une expression de déception pour l'échec. Nous pouvons ainsi nous limiter à un seul écran.

Maintenant que tout est défini, asseyons-nous devant la machine et construisons cet exercice. Cliquer deux fois sur l'icône de "Auteur !" est la première étape. Vous êtes directement dans le mode de conception.

6.3. Passons sur la machine (un exercice mono-écran)....

Nous voilà devant l'écran de conception d' "Auteur !". C'est d'ici que nous allons, ensemble, construire l'exercice de l'intrus que nous avons conçu dans notre tête et sur papier. Mais avant cela, nous allons présenter les différents concepts de l'interface et leur terminologie. Cette section sera très brève. Pour plus de renseignements sur les différentes fonctions du logiciel, l'utilisateur pourra toujours se référer au mode d'emploi fourni en annexe.

6.3.1. Une brève terminologie

Avant de présenter les différents outils que nous avons mis à la disposition de l'utilisateur, il convient de fixer les idées sur un certain nombre de points de vocabulaire. Ainsi, il n'est peut-être pas évident pour tout le monde de savoir ce qu'est un menu déroulant, une barre d'état, une barre d'outils,... Aussi, nous avons décidé de placer ici une sorte de petit lexique auquel l'utilisateur peut se référer.

Il faut d'abord savoir que tous les programmes Windows contiennent plus ou moins les mêmes éléments. Ces éléments font partie de ce que les concepteurs d'applications appellent la CUA (*Common User Access*). Il s'agit d'un ensemble de règles qui permettent à un utilisateur de toujours s'y retrouver quand il utilise un programme tournant sous Windows. C'est la CUA qui définit la place du menu, la place des éléments dans le menu, et bien d'autres choses encore.

Premier élément : tout programme tournant sous Windows est contenu dans une fenêtre (*Main Window* en anglais). C'est dans cette fenêtre que les événements sont pris en compte pour cette application. Un clic en dehors de cette fenêtre provoque un changement d'application.

Deuxième élément en commun dans toutes les applications, la barre de titre (appelée *Caption Bar* par les anglophones). Cette barre contient normalement le nom du programme utilisé. Dans notre cas, cette barre de titre contient le nom du programme "Auteur !". Par défaut, sous Windows, cette barre est bleue. Mais il se peut très bien qu'elle soit d'une autre couleur (c'est selon les goûts de l'utilisateur). Aux extrémités de cette barre de titre, il y a, à gauche, la case système et à droite les boutons de réduction et de maximisation de la fenêtre principale de l'application.

Troisième et dernier élément en commun, tout programme Windows est dirigé à partir d'un menu. Celui-ci se trouve juste sous la barre de titre. Il existe des règles très strictes quant à la création d'un menu, nous ne les citerons pas ici, car ce n'est pas le propos de notre mémoire. Remarquons toutefois qu'il y a presque toujours un menu Fichier qui se trouve à gauche, et un menu Aide qui se trouve à droite. Les autres menus déroulants sont facultatifs, mais les deux menus cités sont quasi obligatoires.

Les éléments que nous allons présenter maintenant, ne sont pas des éléments communs à toutes les applications Windows. Ce sont des éléments qui sont définis par la CUA mais qui apparaissent ou n'apparaissent pas selon l'humeur ou le goût du concepteur de l'application. Il y a aussi des questions de mode qui interviennent (ainsi la barre d'outils est très à la mode pour le moment. On la retrouve un peu partout : dans les traitements de texte, les tableurs, les compilateurs,...).

Le premier de ces éléments dont nous allons parler est la barre d'état (*Status Bar* en anglais). La barre d'état est cette ligne, généralement grise, située au bas de la fenêtre principale de l'application. Elle sert généralement à fournir des indications rapides sur les actions entreprises par l'utilisateur. Ainsi, celui-ci n'a plus besoin de recourir à l'aide en ligne pour les concepts les plus simples du programme. Ceux-ci lui sont expliqués en une ligne au bas de l'écran.

Ensuite, il y a les ascenseurs. Les ascenseurs sont deux barres (une verticale et une horizontale) munies, à leurs extrémités de flèches. Ces flèches permettent d'atteindre des parties de l'écran qui ne sont pas visibles. Ainsi dans "Auteur !" ces deux ascenseurs permettent d'atteindre les parties de l'écran situées sous le menu ou sous la barre d'outils et la barre d'état (ce n'est pas grand chose en plus, mais c'est utile). Entre les deux flèches d'extrémité, il y a une case qui indique où on se trouve dans l'écran.

Enfin, dernier élément d'interface que nous présenterons dans ce lexique, la barre d'outils. Aussi appelé ruban d'outils, il s'agit d'une zone de l'écran qui permet de faire la même chose que certains items du menu mais d'une manière plus rapide et surtout, sans devoir retenir soit où cet item se trouve dans les menus, soit la combinaison de touche pour y arriver. Avec le ruban, un clic de la souris suffit.

Nous employons souvent les termes "clic" et "double clic". Le terme "clic" est le terme qui signifie que l'utilisateur appuie une seule fois sur le bouton de la souris (par défaut le bouton gauche). Le terme "double clic" signifie que l'utilisateur appuie deux fois très rapidement sur le bouton de la souris (toujours celui de gauche). Le double clic correspond donc à deux clics très rapprochés.

Enfin, et pour fixer définitivement les idées, nous allons définir deux concepts propres à "Auteur !" (en fait ce sont deux concepts de Windows utilisés par "Auteur !"). Ces deux concepts sont ceux de fichier-image et fichier-son. Le fichier-image est un fichier (donc un espace réservé et alloué sur le disque dur ou sur une disquette) qui contient une image au format BITMAP. Ce format défini par Windows est un format très pratique à utiliser et à programmer. Lors de l'installation normale de "Auteur !", une série de fichiers-images est fournie. L'utilisateur peut piocher parmi ces images pour définir les exercices. Il va de soi qu'il peut aussi en ajouter dans la collection (pour savoir comment faire, l'utilisateur intéressé consultera le mode d'emploi fourni en annexe). Le concept de fichier-son est le même. Il ne s'agit plus d'image mais d'un son au format WAVEFORM défini par Windows. Encore une fois c'est ce standard qui a été choisi parce qu'il est relativement simple à utiliser et à programmer. L'utilisateur qui voudrait savoir comment ajouter des fichiers-sons à la collection déjà fournie lors de l'installation normale de "Auteur !" consultera le mode d'emploi fourni en annexe.

Ceci termine le petit lexique que nous avons voulu établir afin que les idées soient claires. Nous allons maintenant expliquer plus en détail de quels outils l'utilisateur dispose pour travailler avec "Auteur !".

6.3.2. Présentation des outils

Dans le mode de conception, l'utilisateur a devant lui un écran vide sur lequel il peut ajouter des éléments. Il y a également un menu (situé juste sous la barre de titre du programme), ce que l'on appelle une barre d'état, destinée à donner des renseignements sur les différentes actions entreprises par l'utilisateur (elle se trouve en bas de l'écran), et enfin, il y a une barre d'outils. Selon les options que vous avez choisies, cette barre d'outils se trouve soit à gauche de l'écran, soit au-dessus de l'écran. Il se peut aussi qu'elle soit invisible. Dans ce dernier cas, le menu "Insère" peut servir de palliatif. Maintenant que les éléments de l'interface sont définis, il nous reste à présenter les différents outils. C'est l'objet de la figure suivante.



Figure 6.1. : Les outils du logiciel "Auteur !"

Reprenons ces boutons un par un :

- **Le chargement d'un exercice complet** : l'utilisateur qui fouille un peu les menus du logiciel aura remarqué la possibilité de charger et de sauvegarder soit un exercice complet, soit un seul écran. Les deux premiers boutons permettent de charger (1^{er} bouton) et de sauvegarder (2^e bouton) un exercice complet. Pour charger et sauvegarder un écran seul, il faut nécessairement passer par les menus;
- **Le sauvetage d'un exercice complet** : c'est l'opération inverse de celle du premier bouton. Cet outil permet de sauver le travail déjà effectué lors de la création d'un exercice. C'est un outil destiné à faire gagner du temps à l'utilisateur. En effet, il ne devra pas refaire tous les exercices pour chaque enfant. Il lui suffit de créer un exercice de base et de le moduler en fonction des possibilités de l'enfant à qui il le destine;

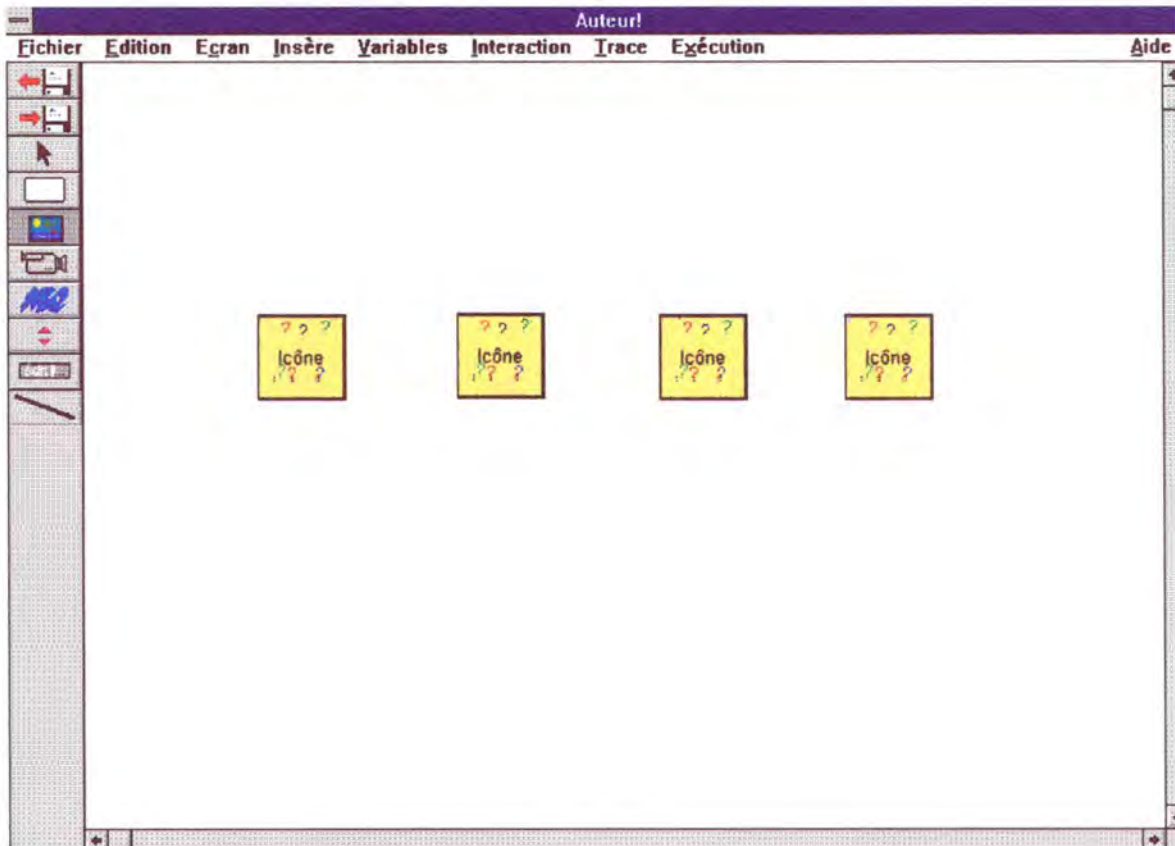
- **L'outil de sélection** : cet outil permet de sélectionner les éléments déjà placés sur l'écran. Il est nécessaire de repasser par cet outil pour faire toutes les modifications sur un élément. En effet, pour modifier un élément de l'écran, il faut le sélectionner et cliquer deux fois dessus. Cela ne peut se faire qu'à partir de l'outil de sélection;
- **Création d'un nouvel écran** : pour créer un exercice multi-écrans (voir plus loin), il faut passer par cet outil. C'est un outil étrange car il donne l'impression que rien ne se passe, s'il n'y a rien sur l'écran courant (celui où l'utilisateur se trouve). Un clic sur cet outil a simplement pour effet de créer un nouvel écran devant l'écran courant. Il donne l'impression, si l'utilisateur a positionné des éléments sur l'écran courant, que ceux-ci sont effacés. Ce n'est évidemment pas le cas. Simplement le programme considère que si l'utilisateur choisit de créer un nouvel écran, il en finit avec l'écran courant. Il cache donc l'écran courant et le remplace par un écran vide où l'utilisateur peut continuer son travail;
- **Création d'une nouvelle icône** : c'est certainement l'outil le plus utilisé par l'utilisateur. C'est lui qui permet d'ajouter des éléments sur l'écran. Ces éléments, les icônes, sont paramétrables par un simple double clic;
- **Création d'une nouvelle animation** : l'animation est un outil de renforcement puissant. Il permet de définir une brève séquence animée à la façon d'un dessin animé. L'animation peut être répétée plusieurs fois pour donner une impression de film;
- **Création d'un texte** : le texte sert quasi exclusivement pour créer des exercices à l'attention des enfants qui savent déjà lire. Il peut aussi servir à l'éducateur qui crée l'exercice pour mettre un commentaire destiné à un autre éducateur ou un commentaire pour savoir où il en est dans sa création;
- **Création d'un défileur** : le défileur était prévu, au départ, pour éviter que l'écran soit surchargé. Nous n'avons pas poussé plus en avant nos investigations sur cet outil ;
- **Création d'un champ d'édition** : le champ d'édition n'a de sens que si l'implémentation des variables est faite. Or nous n'avons pas eu le temps de terminer cette gestion. Le champ d'édition est donc présent dans la barre d'outil, mais il n'est pas activable. (N.B. : la remarque est la même pour le défileur);
- **Création d'une ligne de séparation** : la ligne de séparation permet de définir différentes parties dans l'écran. Comme les autres éléments affichables à l'écran, il suffit d'un double clic pour changer les attributs de cette ligne (épaisseur du trait, couleur,...);
- **Création d'un circuit** : le circuit est un élément un peu spécial. Il y a certaines valeurs fixées par défaut. Il faut nécessairement les adapter pour pouvoir s'en servir. Les différentes largeurs des zones du circuit, les actions associées à chaque zone,... Il faut tout définir. Le circuit se différencie des autres éléments dans la mesure où il faut cliquer sur les autres éléments pour les sélectionner quand on est en mode "utilisation". Le simple fait de changer de zone suffit à déclencher des

actions lorsqu'on travaille avec des circuits. C'est la grosse différence avec les autres outils.

6.3.3. Conception de l'exercice sur machine

Pour concevoir cet exercice de l'intrus sur la machine, nous allons procéder en trois étapes.

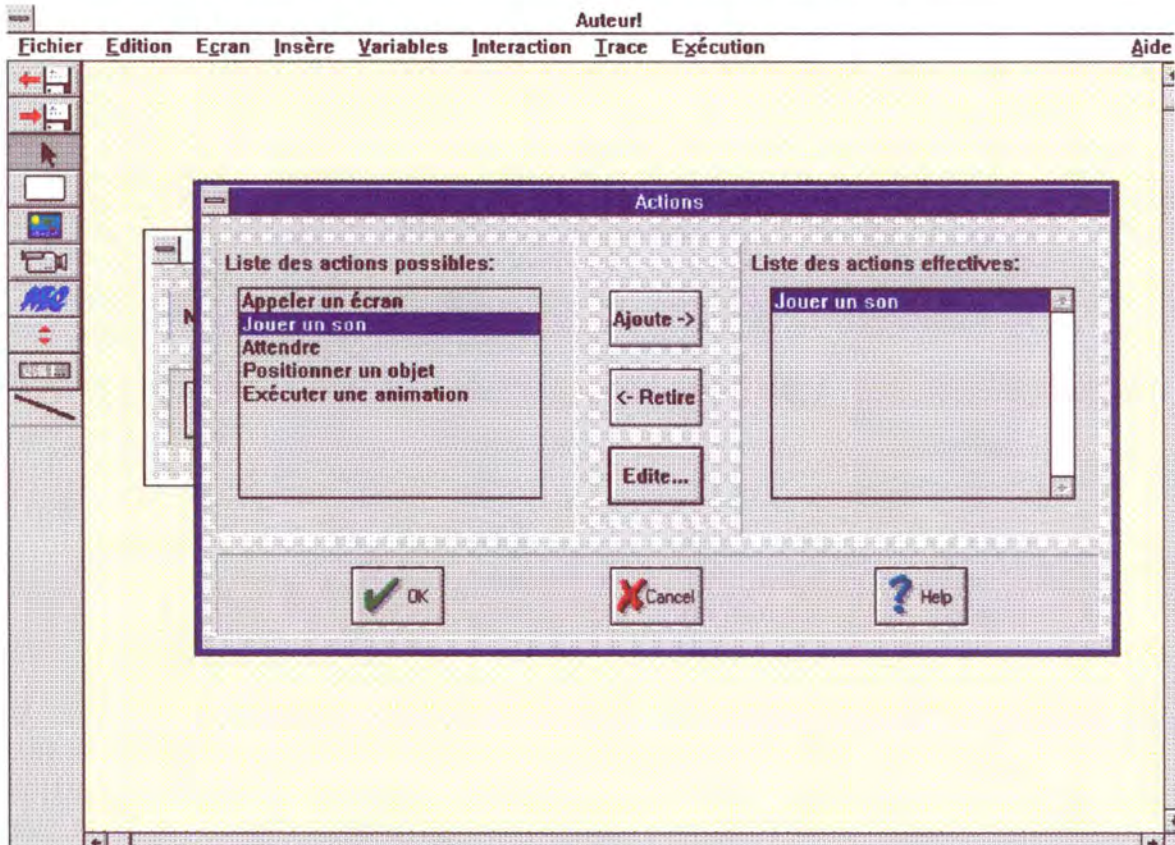
La première de ces étapes est de placer 4 icônes. Pour cela, l'utilisateur sélectionne l'outil "icône". C'est celui avec un petit dessin représentant une île perdue sur un océan et un petit soleil en haut à gauche. Le curseur de la souris change alors et prend la forme de l'outil. L'utilisateur place 4 icônes "vierges" en ligne, comme prévu dans la version papier de l'exercice. Il doit obtenir quelque chose dans le genre de ce qui suit :



Les quatre carrés jaunes sont les quatre futurs pictogrammes. Pour le moment, ils n'affichent pas encore les images que nous avons décidé d'y mettre. C'est normal, car lorsque l'on crée une icône, celle-ci est affichée avec un pictogramme par défaut. Nous changerons ces pictogrammes lors d'une étape ultérieure.

La seconde étape va nous faire modifier le fond d'écran. Nous allons en profiter pour adjoindre une action à l'écran. Il devra jouer un son qui reprend le but de l'exercice. Pour faire ces modifications, l'utilisateur sélectionne la flèche dans la barre d'outils. Il faut cliquer deux fois sur le fond. Le concepteur choisit le bouton "Couleur" et donne à l'écran la couleur qu'il souhaite pour le fond de l'exercice (il ne faut pas oublier que nous avons défini une couleur neutre sur papier). Une fois que cela est fait, il choisit le bouton "Action". Dans la boîte de dialogue qui s'ouvre, il sélectionne l'action "Jouer un son" puis appuie sur le bouton "Ajoute ->" et ensuite sur le bouton "Edite". Il choisit le fichier qui correspond à l'explication du jeu, puis clique sur "Ok".

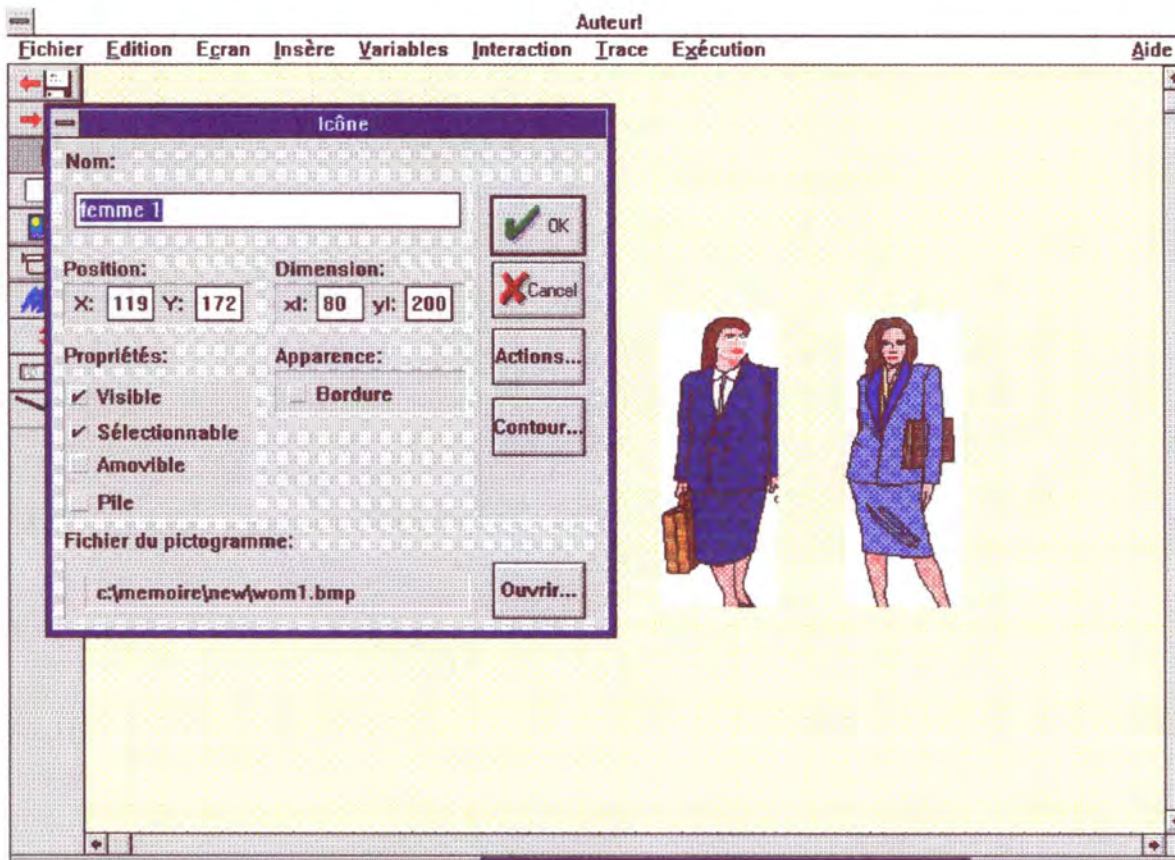
En appuyant sur "Ok" l'utilisateur revient à la situation suivante :



Nous remarquons que le fond d'écran est déjà dans la couleur choisie. Maintenant que nous avons changé les paramètres de l'écran, nous allons pouvoir changer les paramètres des icônes. Au cours de cette troisième étape, nous allons pouvoir changer le dessin et associer une action à chaque icône. Pour des raisons évidentes de répétitions inutiles, nous ne ferons la manoeuvre qu'une seule fois.

Notons aussi que l'on peut changer le nom de l'écran et lui donner un nom plus significatif ; "Intrus 1" est plus significatif que "Ecran 1". Au moins, comme ça, on sait de quoi on parle.

La troisième étape débute par un double click sur une des icônes. La boîte de dialogue suivante apparaît alors :



Nous proposons de renommer les icônes : ainsi, nous aurons les icônes "Femme 1", "Femme 2", "Femme 3", "Homme 1". Nous en sommes à l'icône "Femme 1". Une fois qu'elle sera renommée, il faut cliquer sur le bouton "Actions...". Une nouvelle boîte de dialogue apparaît. Comme pour l'écran, on choisit l'action "Jouer un son". Il faut éditer cette action après l'avoir ajoutée dans la liste. N'oublions pas que l'intrus n'est pas cette icône. Donc si l'enfant clique sur "Femme 1", il devrait entendre un son exprimant la déception. Il faut donc bien choisir le son. Quand le choix sera fait, il faut appuyer deux fois sur les boutons "Ok". On se retrouve dans la boîte de dialogue d'édition d'une icône. Une fois là, il faut appuyer sur le bouton "Ouvrir..." et choisir le pictogramme dont le nom est "WOM1.BMP". Il faut appuyer sur "Ok", et encore une fois sur "Ok".

Si cette suite d'action est répétée pour toutes les icônes respectivement avec les fichiers "WOM2.BMP" pour l'icône "Femme2", "WOM3.BMP" pour l'icône "Femme 3" et "MAN1.BMP" pour l'icône "Homme 1", on obtient, à peu de choses près, l'écran suivant (Attention, il ne faut pas oublier que le son pour l'icône "Homme 1" doit être un son exprimant la réussite) :



Il ne reste plus qu'à tester l'exercice. Pour cela, on choisit l'option "Exécution" du menu. Ce dernier ainsi que la barre d'outils disparaissent. L'exercice se fera en plein écran. On peut essayer l'exercice que nous venons de créer : un click sur une icône représentant une femme doit amener un son exprimant la déception, le click sur l'icône représentant un homme doit amener un son exprimant la réussite.

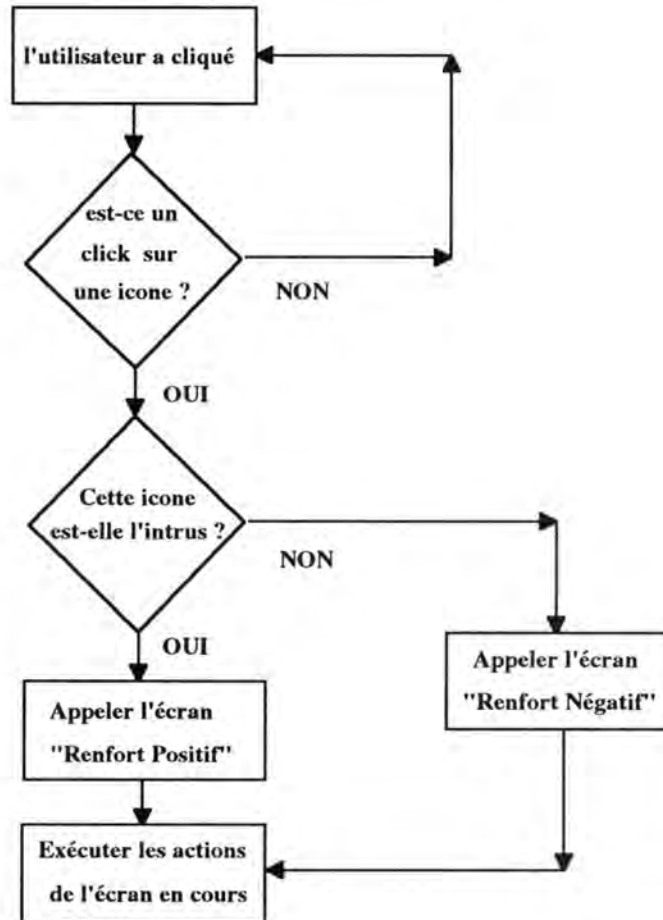
6.4. A vos crayons (un exercice multi-écrans)...

Maintenant que nous maîtrisons bien la technique de création d'un exercice sur un seul écran, voyons comment nous allons pouvoir ajouter des écrans supplémentaires. Ces écrans, sur papier, serviront de renforcements positifs et négatifs. Nous n'allons donc ajouter que deux écrans. L'un sera de couleur verte et reprendra le son exprimant la réussite. Il affichera un pictogramme montrant à l'enfant qu'il a réussi l'exercice. Pour éviter à l'enfant des manipulations inutiles, le retour à l'écran principal de l'exercice se fera automatiquement après 5 - 10 secondes. L'autre écran sera de couleur rouge, il comportera aussi un dessin montrant à l'enfant que sa tentative a échoué. Il reprendra également le son exprimant l'échec, la déception. De même que pour l'écran de renforcement positif, le retour à l'écran principal du jeu se fera automatiquement.

Nous appellerons le premier de ces écrans "Renfort Positif" et le second "Renfort Négatif". Ainsi, nous ne pourrions les confondre. L'écran "Renfort Négatif" servira dans les trois cas où l'enfant clique sur la mauvaise icône.

Nous pensons qu'ainsi, l'exercice est parfaitement défini. De plus, il est très général car il suffit de changer les pictogrammes dans l'écran principal pour avoir un nouvel exercice.

Sur papier, l'exercice peut se représenter comme suit :



Voyons ce que cela donne sur machine.

6.5. Passons sur machine (un exercice multi-écrans)...

Ici aussi, nous allons travailler par étape. Nous repartons pour cela du résultat de la dernière étape de la conception d'un exercice mono-écran. La première chose à faire, est d'éditer (ie afficher chaque icône avec ses paramètres de manière à pouvoir modifier ces derniers) chaque icône, de manière à enlever l'action "Jouer un son". Nous allons remplacer ces actions par d'autres, mais cela doit se faire plus tard.

Ensuite, quand plus aucune icône ne possède d'action, il faut définir les nouveaux écrans. Pour créer un nouvel écran, nous choisissons le petit rectangle blanc dans la barre d'outils.

Pour créer les deux écrans de renforcement, nous allons devoir cliquer deux fois sur cet outil. Pour vérifier que la création s'est bien faite, nous allons dans le menu et nous choisissons l'élément "Ecran". Si tout s'est bien passé (et il n'y a aucune raison que cela ne se passe pas bien), il doit y avoir trois noms d'écrans dans ce menu déroulant :

- ♦ "Intrus 1" qui est le nom de votre premier écran.
- ♦ "Ecran 2" qui résulte du premier click sur l'outil écran.
- ♦ "Ecran 3" qui résulte du second click sur l'outil écran.

Avant de passer sur un autre écran, nous allons compléter l'écran "Intrus 1" en assignant des actions aux icônes. Un double clic sur chacune des icônes, et un appui sur le bouton "Action" pour chacune d'elles. La colonne de droite de la boîte de dialogue "Action" est normalement vide puisque nous avons effacé les actions "Jouer un son". Nous allons donc choisir, dans la colonne de gauche, l'action "Appeler un écran". Ce sera la seule action que nous assignerons à une icône.

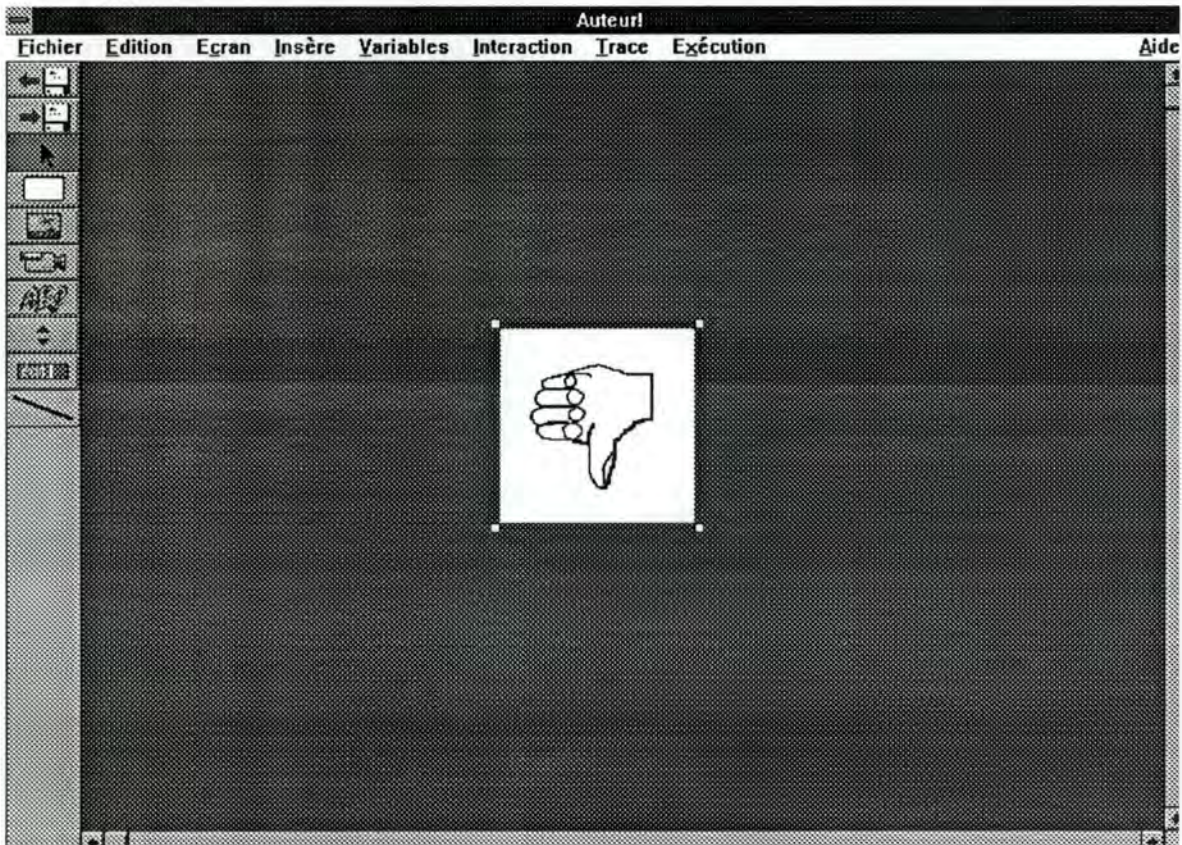
Nous allons maintenant passer à la modification des deux autres écrans. Dans le menu "Ecran", nous cliquons sur l'élément "Ecran 2". Un écran vide doit apparaître. Sur cet écran vide, nous plaçons une icône au milieu. Double-cliquons sur l'icône afin d'en changer le pictogramme. N'oublions pas que nous avons décidé de placer des pictogrammes exprimant soit la réussite, soit l'échec de l'exercice.

Une fois cette modification faite, nous appuyons sur "Ok". Notre icône est placée. Supposons que l'écran "Ecran2" sera l'écran de renforcement négatif. Il faut maintenant changer la couleur de fond de manière à l'amener au rouge. Il faut aussi définir des actions pour l'écran.

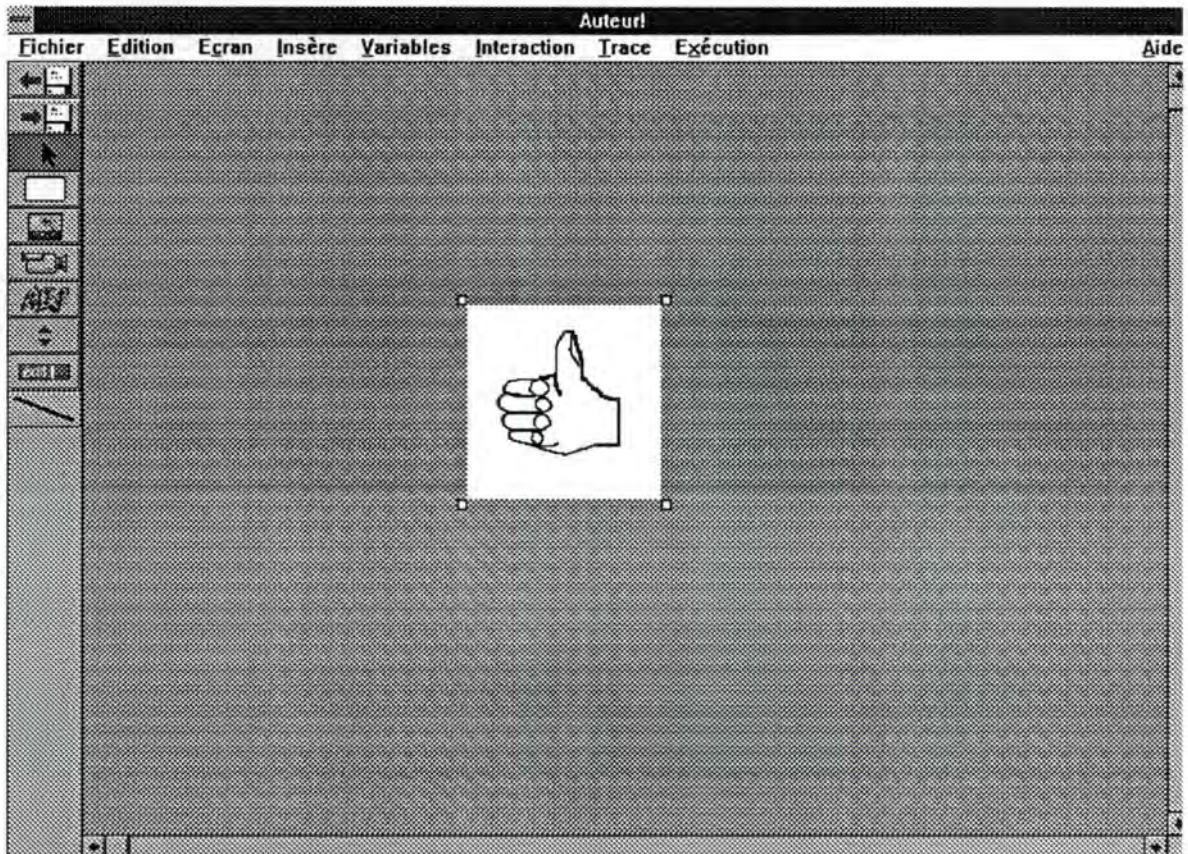
Voici ce que nous nous proposons de faire : Sélectionner la flèche dans la barre d'outils. Double-cliquer sur le fond de l'écran, choisir le bouton "Couleur". Nous choisissons la couleur rouge. Nous appuyons sur "Ok". Nous sélectionnons ensuite le bouton "Action". Dans la colonne de gauche, nous choisissons "Jouer un son" et nous l'ajoutons à la liste. Nous choisissons l'action "Attendre" et nous l'ajoutons à la liste. Nous éditons le son de manière à pouvoir reprendre le son que nous avons utilisé précédemment. Nous éditons l'action "Attendre". Nous spécifions un délai d'attente (dans ce cas, 7 secondes). Les deux actions seront effectuées l'une après l'autre.

N'oublions pas de renommer notre écran : son nom devra être "Renfort Négatif".

Nous pouvons procéder de la même façon pour l'autre écran, celui de renforcement positif. Donc en plus de l'écran principal de l'exercice, nous avons deux écrans ressemblants à ceux-ci :



Ceci étant l'écran "Renfort Négatif". L'écran "Renfort Positif" est le suivant : (voir page suivante).



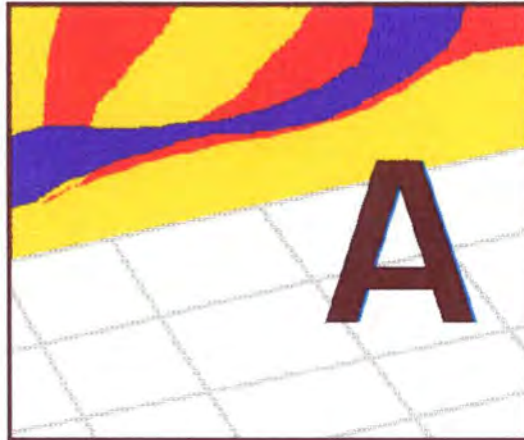
Voilà notre exercice fin prêt à être exécuté. Attention toutefois : il faut bien veiller à ce que l'enfant comprenne l'exercice. Il se peut par exemple qu'il ne comprenne pas le terme "clic". Il sera peut-être préférable d'utiliser un balayage. Le balayage consiste à faire défiler, à un rythme défini par le concepteur, les différentes icônes. Elles seront sélectionnées tour à tour et la réponse de l'enfant se fera par l'intermédiaire de l'équivalent d'un clic sur le bouton gauche de la souris (on peut brancher un contact unique sur le bouton gauche de la souris).

Vous pouvez évaluer votre exercice à posteriori, en regardant la trace laissée par celui-ci. Il vous suffit d'éditer le fichier "TRACE.TRA". Mais pour cela, vous devrez dire au programme qu'il doit enregistrer les actions de l'utilisateur. Cela se fait par l'intermédiaire de l'élément "Trace" du menu.

6.6. Conclusion

En conclusion de ce chapitre, nous dirions que construire un exercice avec "Auteur!" n'est pas difficile. Il suffit de se familiariser avec certains concepts, certains modes de raisonnement.

Bien sûr, nous n'avons montré, par cet exercice très connu qu'est le jeu de l'intrus, qu'une partie élémentaire de ce qu' "Auteur!" peut faire. Mais les concepts de base sont ainsi posés. L'imagination des éducateurs et des thérapeutes, ainsi que l'utilisation des autres outils feront le reste.



Chapitre 7

Structures de données

Où l'on trouve toutes les structures de données : tant celles déjà présentes au moment de la reprise du logiciel en septembre que celles ajoutées cette année.

7.1. Introduction

Conscient que nous nous adressons à un public mixte de non-informaticiens et d'informaticiens, c'est à ces derniers que ce chapitre s'adresse plus spécialement. Il décrit les objets que nous avons utilisés et les objets que nous avons ajoutés. Nous ferons un bref rappel sur la programmation orientée objet, sur le pourquoi de son emploi. Nous constaterons que dans le cas d' "Auteur!", la programmation orientée objet est vraiment bien adaptée.

7.2. La programmation objet : un outil bien adapté

La programmation orientée objet est un nouveau concept de programmation créé, comme beaucoup de concepts nouveaux en informatique, au PARC (Palo Alto Research Center). C'est là, que la première fois, on a écrit un programme dans un langage orienté objet. Depuis, bien des langages ont adopté ces nouveaux concepts : on peut programmer "en objet" en Pascal, en C++, en Basic,...presque tous les langages offrent maintenant une extension orientée objet (même le cobol). Mais là n'est pas notre propos.

Dans le cas d' "Auteur!", le programme que nous avons reçu en septembre 1993 était déjà écrit dans un langage orienté objet. Nous n'avons fait que reprendre ces concepts et essayé de les comprendre. La programmation orientée objet est un tout autre mode de programmation. Sous Windows, la programmation orientée objet passe par le biais de messages. Ce sont eux qui rythment l'exécution du programme. En fait le plus dur quand on programme dans le paradigme orienté objet, c'est qu'on ne voit plus nettement la structure du programme. On écrit des méthodes répondant à tel ou tel cas de figure, tel ou tel message, et on essaye de prévoir tous les cas de figure possibles. Quand on écrit une structure de donnée en orienté objet, la séquence du programme n'apparaît pas : la première méthode peut très bien être appelée en dernier. C'est une des grandes différences avec la programmation "classique". Les méthodes d'un objet peuvent donc être déclarées pêle-mêle, sans aucune distinction d'ordre d'exécution.

Autre différence, en programmation classique, pour définir une structure de données, on emploie une déclaration de structure : c'est le mot réservé "RECORD" en Pascal, ou le "STRUCT" en C et C++. Les données de ces structures sont publiques par défaut. Ce qui signifie que n'importe qui peut y avoir accès. Le but de l'orientation objet étant de cacher un maximum de choses à l'utilisateur, il ne peut donc être question de laisser n'importe qui accéder aux données d'un objet. En C++, lors d'une déclaration de classe, les données sont déclarées privées par défaut. Il faut que le programmeur dise explicitement qu'il veut que ses données soient publiques. La plupart du temps, les données restent privées et les programmeurs écrivent des méthodes pour accéder à ces données.

Autre avantage de la programmation orientée objet, elle permet de réutiliser du code assez facilement. C'est ce que nous avons fait dans la nouvelle version du programme. En fait, nous avons réutilisé les modules écrits par notre prédécesseur sans trop de problèmes. C'est à peine si nous avons dû modifier certains paramètres.

La réutilisation en objet est donc très facile. Supposons que nous ayons un objet "POINT" (c'est l'exemple classique), nous pouvons réutiliser les données et les méthodes écrites pour cet objet dans un objet "CERCLE". Il nous suffira de dire qu'un cercle part d'un point central. Il nous faudra ajouter une donnée : le rayon du cercle, et une méthode propre au traçage du cercle. Cette méthode de programmation s'appelle l'héritage. C'est un des concepts fondamentaux en programmation orientée objet.

Le dernier avantage sur lequel nous nous attarderons, est la notion de constructeur. En programmation classique, lorsqu'on déclare une structure de données, il faut, avant de l'utiliser, l'initialiser. En programmation orientée objet, lorsqu'on déclare une variable de type objet, le simple fait d'entrer dans le corps de la fonction entraîne l'appel d'une méthode spéciale appelée "constructeur". Ce constructeur initialise, au minimum, toutes les données contenues dans l'objet. Au cas où l'objet fait appel à des instructions de réservation de mémoire, la méthode "destructeur" devra être appelée en fin de fonction ou en fin de programme (pour les variables globales).

Comme nous l'avons vu, la programmation orientée objet est un concept de programmation tout à fait nouveau. Il a de nombreux avantages et le seul inconvénient que nous lui trouvons est en fait qu'il faut du temps pour comprendre comment cela fonctionne. Surtout si l'on est habitué à la programmation classique. Nous voulons aussi signaler que la programmation orientée objet est un des seuls modes valables de programmation sous Windows (avec les 4GL). En effet, la programmation classique est toujours possible mais elle demande beaucoup plus de temps et beaucoup plus de code pour arriver aux mêmes résultats. Sous Windows, la programmation classique se fait en utilisant les API's les unes à la suite des autres. La programmation orientée objet est, elle aussi basée sur les API's, mais la possibilité de réutiliser le code déjà écrit est beaucoup plus grande. Le même programme, par exemple, qui se contente d'écrire une phrase à l'écran, est écrit beaucoup plus simplement dans le paradigme orienté objet que dans le paradigme de la programmation structurée.

Mais pourquoi la programmation orientée objet est-elle un outil bien adapté pour "Auteur !" ? En fait, tout les outils utilisés dans "Auteur !" dérivent d'un seul concept : l'élément. Une icône est un élément, une animation est un élément,... tous les outils héritent de certaines propriétés de l'élément. Les propriétés spécifiques aux outils, telles les méthodes d'affichage, d'édition,..., sont écrites dans leurs objets respectifs.

Nous nous sommes basés sur la structure d'objet déjà existante, et créée par les informaticiens de chez Borland : OWL. OWL (Object Windows Library) est un ensemble d'objets sur lequel peut se greffer n'importe quelle application. Il s'agit d'une structure hiérarchisée : chaque objet du dessous de la hiérarchie hérite des caractéristiques d'un ou plusieurs autres objets. La figure de la page suivante montre une partie de cette hiérarchie. Elle montre aussi les liens entre les objets que nous avons créés pour "Auteur !" et la hiérarchie établie par Borland.

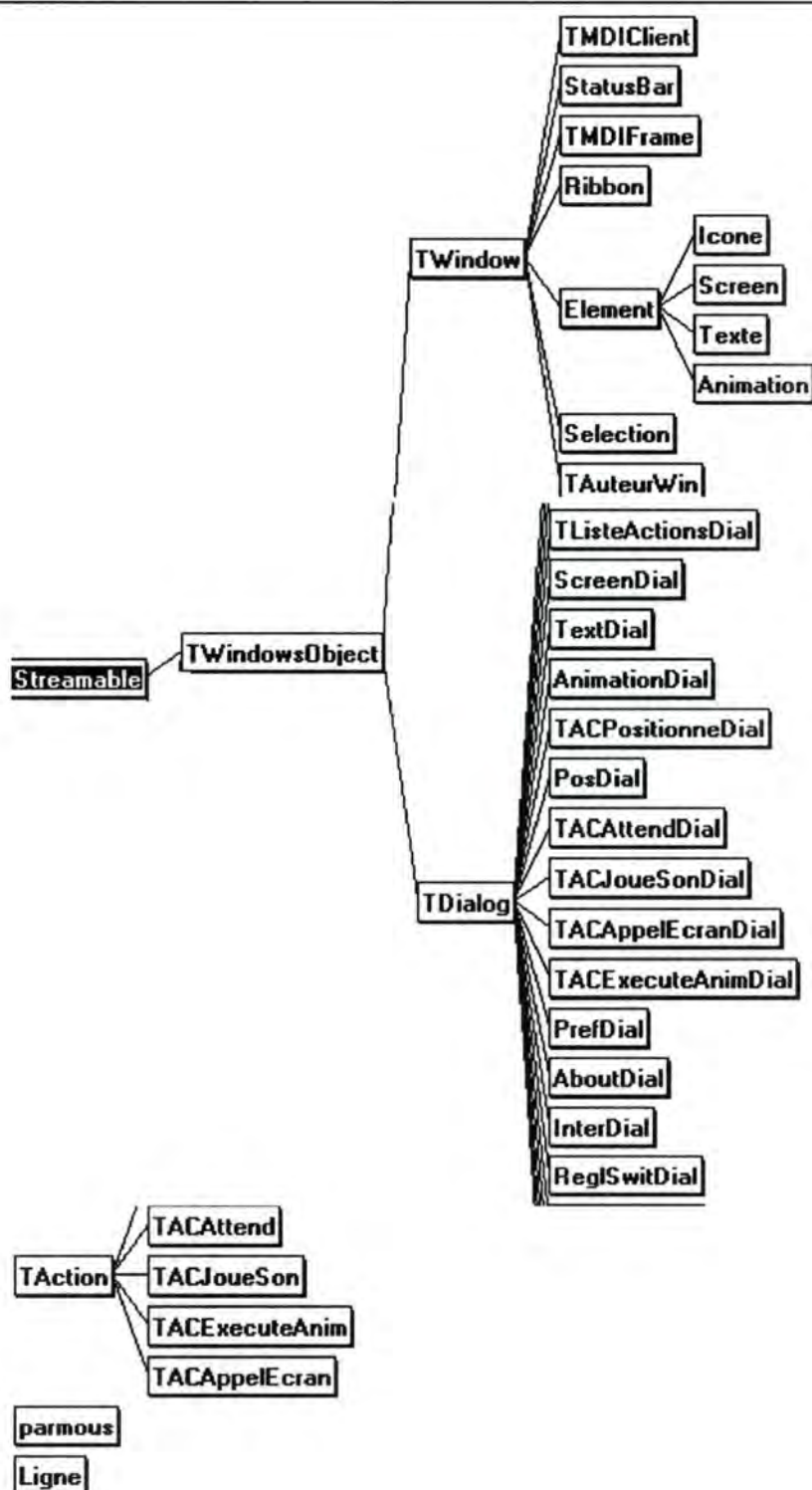


Figure 7.1. : Une idée de la structure en place

Seuls les objets Streamable, TWindowsObject, TWindow, TDialog, TMDIClient, TMDIFrame, tagWP, tagLP étaient déjà écrits. Ce sont les objets de base de OWL. Les autres objets ont été définis soit par Luc Vandenabeele, soit par nos soins.

7.3. Structures de données des éléments préexistants

Pour chaque élément préexistant, nous allons donner une brève description des données et des méthodes. Pour des raisons de facilité, ces objets sont coupés en deux : une partie est l'objet proprement dit, avec ses données et ses méthodes, une autre partie est l'objet de dialogue correspondant. Ce dialogue contient généralement une instance de l'objet en question, pour des raisons de facilité de transfert de données.

Nous représenterons l'objet sous forme graphique avec les conventions suivantes :

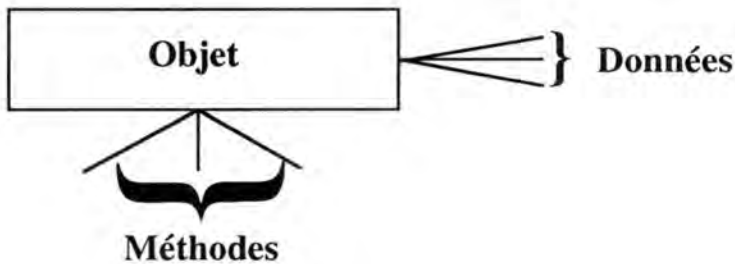


Figure 7.2. : Convention de représentation des structures de données

7.3.1. L'objet "élément"

Un élément est caractérisé par un type, un nom, un identifiant, 4 paramètres de taille et position, un booléen, disant s'il est sélectionné ou non, un indicateur de propriété possédée par l'objet. Il y a aussi une liste d'actions, une sélection (objet qui est en fait une liste d'objets sélectionnés à l'écran) et un état (objet indiquant l'état de sélection de l'élément).

C'est dans cet objet que sont définies la plupart des méthodes utilisées par les autres objets. Les objets "écran", "icône" et "animation" sont des objets dérivés de l'objet "élément".

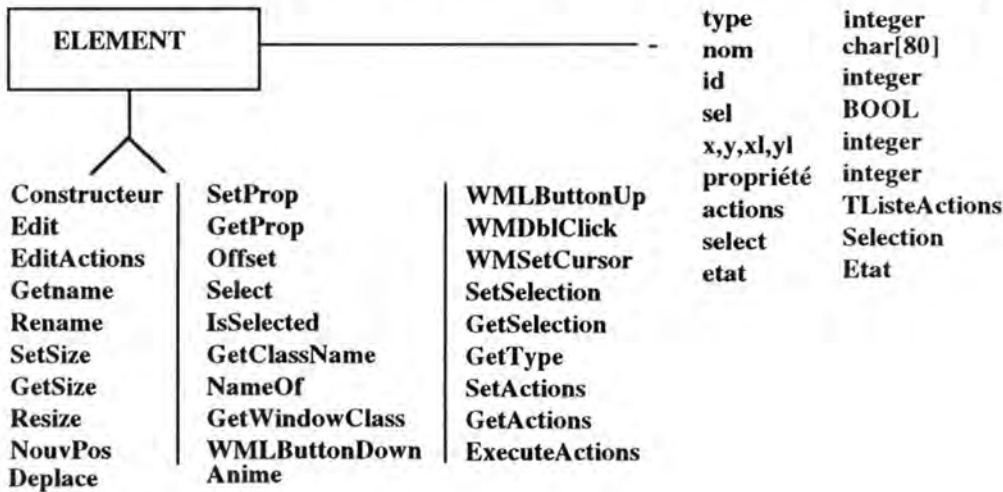


Figure 7.3. : L'objet "élément"

7.3.2. L'objet "écran"

Un écran est caractérisé par un nombre d'icônes, une couleur et un nom. il contient aussi une instance de l'objet sélection et le nom de l'écran précédent dans la liste des écrans. Un écran est un élément de type 1. Il hérite de toutes les méthodes de son objet père et quelques méthodes typiques lui sont ajoutées. Parmi ces dernières, on trouve les actions à faire lorsqu'on clique sur l'écran, lorsqu'on veut éditer l'écran, lorsqu'on veut ajouter des éléments dans l'écran,...

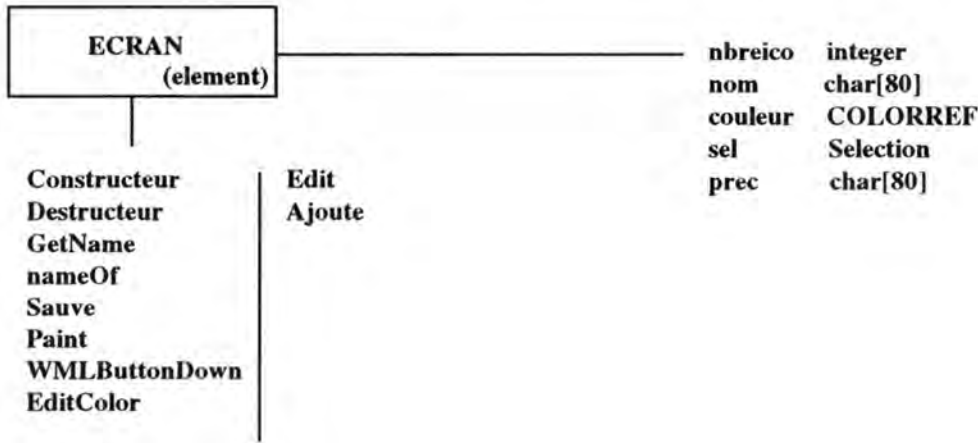


Figure 7.4. : L'objet "écran"

7.3.3. L'objet "icône"

L'objet "icône" est lui aussi un dérivé de la classe "élément". Ses caractéristiques sont un peu plus nombreuses. Parmi celles-ci, on retrouve le nom, le nom du fichier image, la position, la taille, les propriétés,... En plus de ses attributs, l'icône possède une donnée objet qui est en fait un objet "dessin". C'est cet objet qui est responsable des calculs de la taille du dessin,...

Les 2 méthodes principales sont des méthodes d'édition et de sauvetage. Les autres méthodes permettent un accès plus facile aux données privées de l'objet.

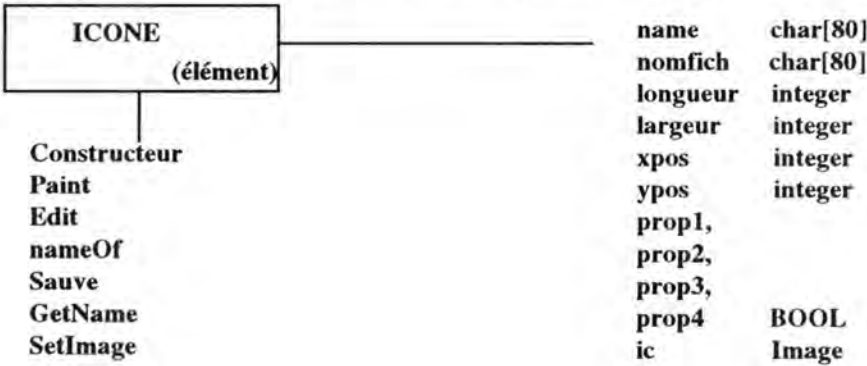


Figure 7.5. : l'objet "icône"

7.3.4. L'objet "animation"

L'objet "animation" est similaire à l'objet "icône". Il possède, à peu de chose près, les mêmes données. La différence réside dans le fait que l'animation ne se compose pas d'une seule image, mais d'un groupe d'images. Ces images sont enchaînées les unes à la suite des autres de manière à créer une sorte de dessin animé.

Les méthodes principales de cet objet sont la sauvegarde, l'édition, l'animation. Ces trois méthodes ne figurent pas dans l'objet "élément". Elles sont spécifiques à l'objet "animation".

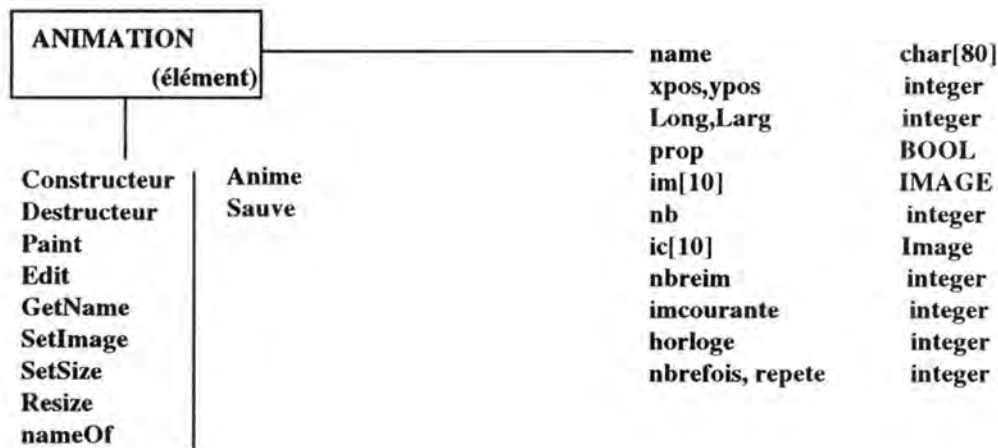


Figure 7.6. : L'objet "animation"

7.3.5. L'objet "TAction"

L'objet "TAction" n'a aucune donnée propre. C'est juste un objet générique servant à définir, par héritage, les actions particulières telles que "Jouer un son", "Appeler un écran",...

Les méthodes de cet objet sont, elles aussi, génériques. Elles ne font donc rien. Elles attendent une spécialisation : chaque action possède son propre mode d'exécution, sa propre méthode de sauvetage, de rechargement.

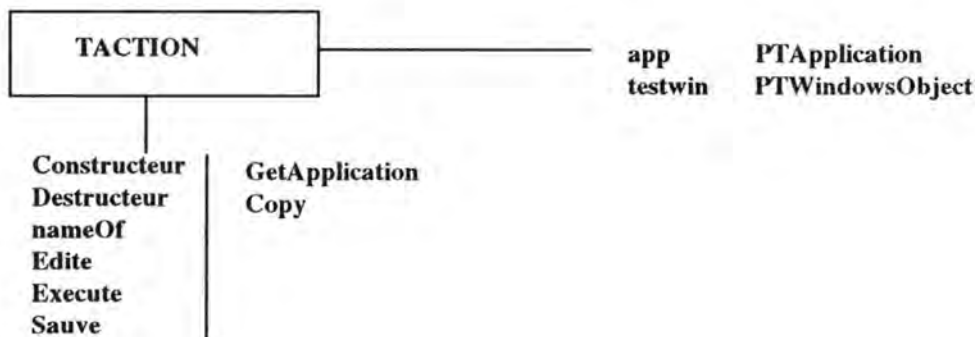


Figure 7.7. : L'objet "TAction"

7.3.6. L'objet "TListeActions"

Une liste d'actions, c'est d'abord une liste de 10 actions maximum. Ces 10 actions sont de type "TAction". Ensuite, il y a un pointeur vers l'élément auquel appartient cette liste d'actions. Enfin, il y a un pointeur vers l'application(usage non connu).

Les méthodes sont essentiellement les méthodes de sauvetage et de rechargement, les méthodes d'exécution et d'édition d'une action. Il y a aussi les méthodes d'insertion et de suppression d'une action dans la liste.

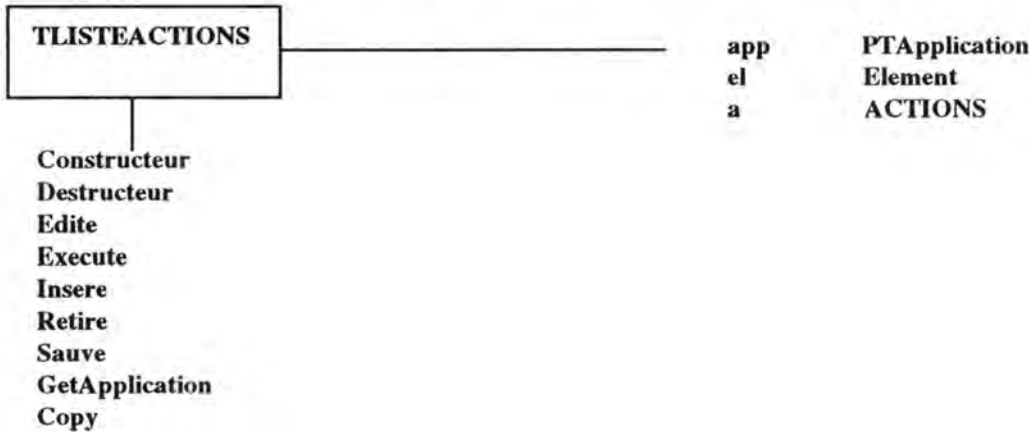


Figure 7.8. : L'objet "TListeActions"

7.4. Structures de données des éléments ajoutés

Trois outils ont été ajoutés cette année : il s'agit du texte, de la ligne et du circuit . Le reste des modifications étaient plus internes au programme préexistant. Au départ, la ligne était un outil qu'il nous avait été demandé d'ajouter pour pouvoir construire des circuits. Nous avons décidé par la suite de créer ces deux concepts indépendamment. Le texte est un outil que nous avons décidé d'ajouter de notre propre chef. Il nous a servi de test pour l'implémentation de la ligne, les deux objets utilisant des méthodes d'affichages similaires.

7.4.1. Le texte

Le texte est un élément qui n'est pas comme les autres. Il s'affiche certes dans une fenêtre Windows, mais cette fenêtre reste invisible. Cela nous a posé un petit problème lors de l'implémentation : cette fenêtre, puisqu'elle était censée être invisible, avait une taille très réduite (nous avons fixé sa taille à zéro). Il va de soi que dans une fenêtre de taille zéro, on ne peut pas afficher de texte.

Le texte possède certains attributs : une couleur, une taille, une forme (spécifiée par la fonte utilisée : Times, Arial, Architect,...). L'objet "texte" possède certaines méthodes. Les principales sont celles de dessin du texte (car les fontes sous Windows sont dessinées à l'écran), et celle d'édition du texte. Le texte reçoit une valeur par défaut : c'est cette valeur qui est affichée lorsqu'on crée une occurrence de l'outil texte (c'est-à-dire lorsqu'on clique sur l'écran après avoir sélectionné l'outil "texte"). Le texte n'est pas sélectionnable. Il est présent à l'écran comme information ou comme commentaire pour l'utilisateur (par exemple, pour expliquer le but de l'exercice).

7.4.2. La ligne

La ligne possède certains attributs tels l'épaisseur et la continuité du trait, la couleur, la visibilité du trait,... La ligne est, comme le texte, affichée dans une fenêtre, qui en fait un dérivé de l'objet "élément". Elle n'est donc pas dessinée directement sur l'écran. La ligne sert surtout à séparer deux parties de l'écran (ou plus). Nous n'avons pas prévu que l'utilisateur puisse lui assigner des actions : le clic sur une ligne est un concept difficile, même pour une personne qui manipule bien la souris. La ligne n'est donc pas sélectionnable.

Les méthodes principales de la ligne sont le dessin de la ligne et son édition.

7.4.3. Le circuit

Le circuit constitue le gros morceau de ce travail de programmation. C'est avec cet outil que le centre "La Famille" désire travailler avant tout. En effet, c'est dans le cadre des exercices sur la latéralité que le circuit s'inscrit.

Un circuit est constitué de trois zones :

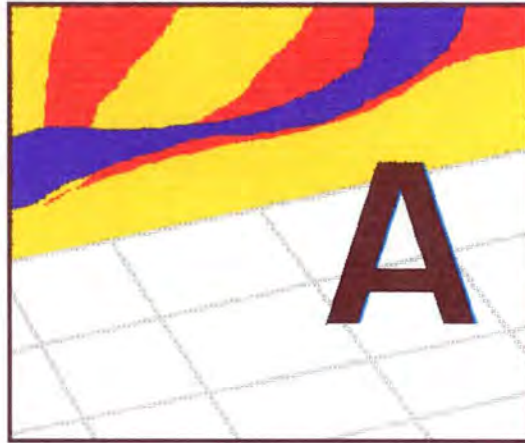
- ♦ La zone de circuit proprement dite : c'est la zone où l'enfant est censé maintenir le curseur;
- ♦ La zone de dégagement : comme sur les vrais circuits pour les formules 1, un dérapage est toujours possible. Nous avons prévu la possibilité de dérapage par la zone de dégagement. C'est donc une zone qui avertit l'utilisateur quand il y entre.
- ♦ La zone de rejet : C'est une zone qui n'existe pas dans la réalité. Quand le curseur pénètre dans cette zone, il est censé, selon le cahier des charges, être rejeté vers le milieu du circuit.

Chacune de ces zones possède, au départ une largeur par défaut, largeur que l'utilisateur peut régler à son gré. De plus, nous laissons l'utilisateur libre de définir une suite d'actions pour chaque zone du circuit. Ainsi, celui qui exécute l'exercice avec le circuit peut entendre un bruit de moteur tant qu'il est dans la zone de circuit, un bruit de dérapage quand il met le curseur dans la zone de dégagement, et un bruit de frein quand il arrive dans la zone de rejet. Les principales méthodes de cet objet sont les méthodes d'affichage et d'édition. Notons également que la gestion des régions par Windows intervient pour une très grande part dans cet objet un peu spécial.

7.5. Conclusion de ce chapitre

Nous avons décidé de rappeler les structures de données existantes parce que notre prédécesseur ne les avait pas expliquées en détail, ce qui nous paraissait dommage. Nous n'avons pas ajouté beaucoup d'outils. Cela est dû au fait que les outils déjà implémentés convenaient plutôt bien pour le genre d'exercices que nous voulions réaliser. Il ne manquait vraiment que la ligne et le circuit pour pouvoir implémenter tous les exercices désirés. Le texte est un petit "plus" que nous avons décidé d'implémenter à titre d'exercice.

Le plus gros de notre travail n'a donc pas été d'ajouter des structures, mais de pouvoir jongler avec les structures existantes (à une exception près) afin de pouvoir créer les exercices demandés.



Chapitre 8

Perspectives

Où l'on découvre ce que l'on pourrait ajouter au programme "Auteur !" pour qu'il soit encore meilleur.

8.1. Introduction

Nous avons voulu placer un chapitre consacré aux perspectives car un programme est toujours améliorable (surtout quand, comme celui-ci, il n'en est qu'à sa deuxième version). Il y a toujours des petites choses que l'on peut rectifier, que l'on peut optimiser. Il y a toujours des éléments nouveaux que l'on veut ajouter...C'est la raison de ce chapitre. Nous espérons bien sûr aller le plus loin possible dans la réalisation des objectifs que nous nous sommes fixés, mais de petites choses pourraient tomber à l'eau au fur et à mesure de l'écoulement du temps. Nous préférons, dans ce dernier cas, signaler les modifications qui pourraient encore être faites.

8.2. Des ajoutes

8.2.1. L'aide en ligne

Nous l'avons déjà signalé à plusieurs reprises dans ce travail, nous n'avons pas eu le temps de réaliser un système d'aide valable. Le seul système d'aide que nous avons essayé de mettre en place est constitué par la barre d'état. Ce concept était déjà présent au départ, mais il n'était pas utilisé. Nous avons donc fait de notre mieux pour que l'utilisateur n'ait pas à recourir à la documentation à chaque manipulation. Malheureusement, ce n'est, à notre avis, pas assez. L'aide en ligne est un outil très facile à utiliser. Il suffit à l'utilisateur de repérer les mots qui l'intéressent. Un simple clic sur ces mots permet l'accès à d'autres mots, et ainsi de suite jusqu'à obtention de l'information voulue.

Ce concept de l'aide en ligne par lien hypertexte n'est pas difficile à réaliser. Windows fournit les outils pour la réaliser : un traitement de texte tel Word for Windows suffit à composer le texte, et le programme Winhelp (fourni en standard avec Windows) est assez facile à manipuler. Les instructions de codage de l'aide ne sont, elles non plus, pas difficiles à maîtriser.

Ce qui est difficile, c'est de composer le texte de manière cohérente. Une aide hypertexte est difficile à composer parce qu'il faut prévoir un plan précis de ce que l'on veut mettre dedans. Mais plus encore qu'un plan, il faut prévoir les enchaînements entre les différentes pages. L'utilisateur peut très bien passer de la page 10 à la page 1 et ensuite passer à la page 67. Il y a tout un jeu de références croisées à gérer.

8.2.2. Les séquence animées

Par le terme "séquences animées", nous entendons les animations issues de capture d'images vidéo. Nous n'oublions pas que nous entrons petit à petit dans l'ère du multimédia. Les captures d'images vidéo, le CD-ROM, les commandes vocales,... sont des concepts avec lesquels il nous faudra compter. Nous estimons que l'imagerie vidéo est très importante, car il s'agit là d'un outil supplémentaire potentiel pour le programme "Auteur !".

Ce serait un outil différent de l'outil "animation" que nous connaissons déjà. Il permettrait au concepteur d'insérer des séquences vidéo dans ces exercices. Imaginons par exemple un exercice où le renfort positif serait constitué par une scène de joie filmée, impliquant l'enfant qui fait l'exercice. L'enfant se verrait sur l'écran. Nous croyons que si l'enfant se voit à l'écran, son attrait pour ce genre d'exercice en sera encore accru.

Evidemment, l'adjonction d'un tel outil augmenterait encore la taille du programme et impliquerait encore des changements assez profonds dans le programme (notamment au niveau des méthodes de sauvegarde et de chargement d'un exercice). Toutefois, Windows (en fait c'est surtout Microsoft) propose un format d'imagerie vidéo : le format AVI. Ce format, au même titre que le format WAVE des fichiers sonores, est assez facile à utiliser. De plus, à la différence des fichiers sonores, ces fichiers ne nécessitent pas de matériel spécifique pour pouvoir être affichés. Il existe en fait des instructions permettant de gérer ces séquences vidéo de manière assez aisée. Et bien que nous n'ayons pas à l'heure actuelle pris connaissance de ces instructions (elles ne sont fournies qu'avec les documentations de programmation multimédia), nous sommes sûr qu'elles sont aussi faciles à utiliser que leurs homologues sonores.

8.2.3. *CD-ROM et commande vocale*

L'emploi du CD-ROM peut être pratique pour le stockage des images et des sons. En effet, une image vidéo, ou un son digitalisé à haute fréquence prennent de la place sur un disque dur. Il serait pratique de pouvoir stocker les images, les séquences vidéos et les sons sur un support externe de grande capacité.

La commande vocale est maintenant fournie, comme nous l'avons signalé, avec certaines cartes sonores (les nouvelles cartes Creative Labs (Sound Blaster AWE32)). Mais il s'agit d'un programme à part, fonctionnant en tâche de fond, et non d'un module que l'on peut inclure dans n'importe quelle application de manière à ce qu'elle réagisse à la voix. Ce programme permet à l'utilisateur d'associer des paroles avec certaines actions que l'on fait souvent avec l'ordinateur. Par exemple, quelqu'un qui utilise beaucoup un traitement de texte peut associer les principales commandes telles "Sauver", "Charger", "Imprimer",... à un son qui une fois prononcé déclenche l'action en question. Il serait pratique de pouvoir avoir cette option dans "Auteur !" : cela permettrait d'ajouter un moyen d'interaction supplémentaire : la voix. Par exemple, les mots "gauche", "droite", "en haut", "en bas" pourraient servir à diriger le curseur et les mots "stop" ou " clic" permettraient de simuler le clic sur un bouton. C'est une "voie" à explorer, si l'on peut dire...

8.2.4. *L'analyse de la trace*

L'analyse de la trace est un autre concept que nous n'avons pas eu le temps de placer dans le logiciel. Il nous avait été demandé, si nous avions le temps, de regarder ce concept d'un peu plus près. L'analyse de la trace telle que définie par Luc Vandenabeele était déjà un premier pas vers une analyse des résultats d'exercices a posteriori. Nous aurions, si nous avions plus de temps, dû faire évoluer l'analyse de la trace pour en faire un concept graphique et non plus un concept textuel.

Si nous observons la trace telle qu'elle existe actuellement, nous constatons qu'il s'agit d'un fichier de texte que les éducateurs responsables peuvent lire à partir de n'importe quel éditeur de texte (qu'il tourne sous DOS ou sous Windows). Notre projet était d'en faire une sorte de programme dans le programme. Ainsi, cette partie du programme aurait pu faire des calculs sur les valeurs présentées dans le fichier texte, et le présenter sous forme d'un ou plusieurs graphiques. De même, pour que les éducateurs et les thérapeutes puissent se faire une idée de la maîtrise des moyens d'interaction de leurs élèves, nous avons envisagé de retracer le parcours effectué par le curseur tout au long de l'exercice. Nous estimons quant à nous qu'une trace réalisée selon les modalités que nous venons d'invoquer serait un pas supplémentaire vers une interface utilisateur encore plus conviviale.

8.3. Des améliorations

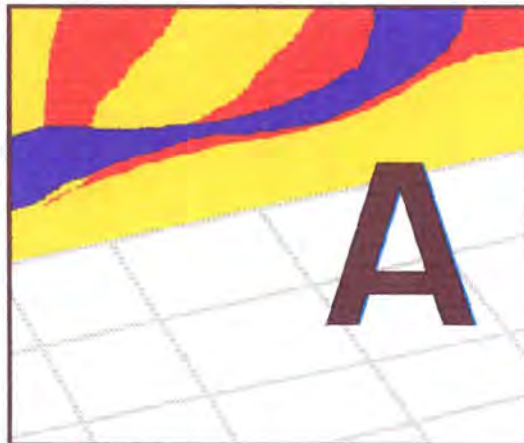
Comme de bien entendu, cette deuxième version d' "Auteur !" est loin d'être à son meilleur niveau d'optimisation. Il reste, de ce côté, pas mal de travail à faire. Il subsiste aussi des petites imperfections : conversions de coordonnées relatives en coordonnées absolues, compatibilité totale entre les écrans, messages qui n'arrivent pas au bon moment ou au bon endroit,...

Ces petits problèmes ont été réglés de manière à ce que le programme puisse quand même tourner sans trop de problèmes. Mais ce sont des imperfections qui, pour un informaticien, sont assez gênantes à présenter. Bien évidemment, ces imperfections ne sont pas graves : elles n'empêchent normalement pas le programme de fonctionner. Nous espérons toutefois qu'elles pourront disparaître au cours de l'histoire de ce logiciel, dont le chemin n'est pas fini. Le travail qui reste donc à faire dans ce domaine est donc un gros travail d'optimisation et de recherche des erreurs restantes.

De même, les outils existants devront, par la suite encore être souvent modifiés pour que les nouveaux concepts que l'on voudra implémenter soient pris en compte. Comme on le voit, il reste encore beaucoup de travail.

8.4. En conclusion de ce chapitre

En conclusion de ce chapitre, nous avons envie de dire que la porte est ouverte. Ouverte à toutes les suggestions. Une partie du travail est déjà faite, une autre partie reste à faire. Le projet étant déjà bien avancé, il reste malgré tout encore beaucoup de choses à faire.



Conclusion

"Auteur !" est un travail de longue haleine. Et bien que non encore terminé, il est déjà opérationnel dans bien des domaines. "Auteur !" est un logiciel qui nous a permis de mieux comprendre certains concepts évoqués tout au long de ces cinq années d'études. En effet, pour réaliser ce programme, nous avons dû mettre en pratique beaucoup de choses que nous avons apprises en théorie.

Et c'est ce qui est formidable dans ce mémoire : la diversité des sujets abordés. Nous avons dû, dans un premier temps, évaluer et comprendre ce que notre prédécesseur avait voulu faire. Mais contrairement à ce que lui avait fait, notre but n'était pas d'évaluer les outils déjà offerts à l'utilisateur. Notre but était de comprendre ce qu'il avait fait de manière à pouvoir réutiliser au mieux le code déjà écrit. Il va sans dire que dans ce domaine, son aide et sa connaissance de l'environnement Windows nous ont été plus que précieuses.

Notre décision d'utiliser "Auteur !" plutôt que de construire un logiciel dédié aux problèmes humains que nous avons rencontrés cette année a été guidé par le fait que le projet de cette année entrerait bien sous la forme d'un nouvel outil pour le logiciel "Auteur !". C'est pour cette raison que nous avons, dès le départ, décidé de travailler avec ce programme.

Un autre domaine abordé par le projet : la maintenance. Il fallait, comme nous l'avons signalé ci-dessus, que nous comprenions parfaitement le travail déjà fait de manière à, le cas échéant, pouvoir être à même de corriger les imperfections et erreurs éventuelles. La maintenance impliquait aussi que nous ajoutions certains éléments tels la sauvegarde et le chargement sans lesquels le programme n'était pas utilisable.

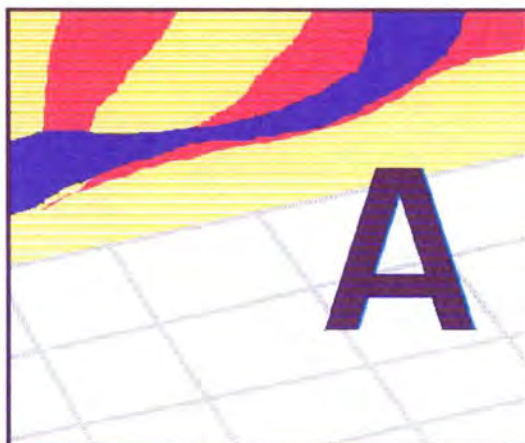
Nous avons, comme notre prédécesseur (mais avions-nous vraiment le choix, à partir du moment où une partie du programme était déjà écrite), utilisé une méthode orientée objet. Toutefois, nous n'avons pas utilisé OBLOG étant donné que les outils que nous voulions ajouter n'étaient pas légion. Notons au passage que nous avons gardé les critères de réutilisabilité du code du logiciel : bien des modules pourraient être réutilisés dans d'autres applications (notamment la gestion du ruban d'outil, le concept d'élément et tout ce qu'il implique, l'icône,...).

La grande nouveauté pour nous fut sans conteste de devoir mener un projet à son terme dans un langage et sous un environnement qui ne nous étaient pas familiers (la programmation Windows en C++ n'est pas une chose aisée, néanmoins, au terme de ce mémoire, ses concepts nous apparaissent beaucoup plus clairs qu'au début).

Nous laissons le programme ouvert pour des modifications ultérieures. Non pas que ces modifications soient obligatoires, mais certaines pourraient augmenter l'attrait des utilisateurs pour ce logiciel. Ces modifications ont fait l'objet du chapitre précédent. Nous pensons sincèrement qu' "Auteur !" ne doit pas s'arrêter à un simple travail de mémoire. Les bases en sont jetées, il serait dommage de ne pas profiter de ces fondations.

Le dernier point auquel nous avons touché durant ce travail, et il n'est pas des moindres, c'est le contact avec l'utilisateur. Cela comprend aussi bien les contacts avec les gens du centre "La Famille" que l'interface utilisateurs ou la population à qui "Auteur !" est destiné. Nous avons déjà signalé les problèmes que nous étions censés traiter cette année. C'est maintenant aux utilisateurs de nous dire si nous avons atteint nos objectifs.

Nous terminerons en disant que nous sommes contents d'avoir pu travailler au service de personnes moins favorisées. Cela nous a fait prendre conscience de notre rôle vis-à-vis de ceux à qui l'ordinateur peut tant apporter. Comme nous l'avons signalé dans le chapitre 2, étant donné la place qu'est destinée à prendre l'informatique dans la vie future, il est normal que nous nous mettions au service de ces gens afin qu'il ne soient pas exclus du monde futur.



Bibliographie

- [Bern, 92] , Claudy Bernard, *Evaluation d'ergonomie d'écran pour logiciels destinés à des enfants Infirmes Moteurs Cérébraux*, Mémoire de Licence et Maîtrise en Informatique, Institut d'Informatique, Septembre 1992.
- [Borl,91] , X., *Borland Languages Help Compiler*, Borland, Scott Valley, 1991.
- [Brou,90] , L. Brousmiche, X. Gillet, *Logiciel de simulation d'un jeu de marionnettes pour enfants infirmes moteurs cérébraux*, Mémoire de Licence et Maîtrise en Informatique, Institut d'Informatique, Septembre 1990.
- [Card, 93] , G Le Cardinal, F. de Barbot, R.M. Appuijojon, *Utilisation de l'informatique pour l'évaluation d'efficiences intellectuelles*, Rapport Hôpital Kremlin-Bicêtre, Paris, France, 1993.
- [Char, 92] , C Charrière, *Ergolab, dans le domaine de l'infirmité motrice cérébrale*, Réunion Hantépsycom, Bütgenbach, 17-19mars 1994.
- [Cong,92] , James L. Conger, *Windows API Bible, The definitive programmer's reference*, The Wait Group Press, Mill Valley, California, 1992.
- [Depl,89] , M. Deplechin, G. Strappazzon, *Un logiciel des gestion des comptes pour personnes handicapées mentales adultes, développement et évaluation*, Mémoire de Licence et Maîtrise en Informatique, Institut d'Informatique, Septembre 1989.
- [Demo,92] , R.Démo, S. Baudrenghien, *Elaboration d'une boîte à outils d'aide à la réalisation d'interfaces pour personnes handicapées*, Mémoire de Licence et Maîtrise en Informatique, Institut d'Informatique, Septembre 1992.
- [Dila,92] , Paul Dilascia, *Windows ++ : Writing Reusable Windows Code in C++*, Addison-Wesley, Readings, Massachusetts, USA, 1992.
- [Howa,91] , J.R. Howard, J.C. Busch, J.A. Watson, D.D. Shade, *The Change-Over to Computer-Based Technology in Early Childhood Special Education*, Journal of Research on Computing in Education, Summer 1991, volume 23 number 4.
- [Lebl,93] , Gérard Leblanc, *Turbo/Borland C++*, Eyrolles, Paris, France, 1993.
- [Lebl,92] , Gérard Leblanc, *Programmation Windows en Turbo C++ et Borland C++*, Eyrolles, Paris, France, 1992.

-
- [Lep,90] ,T. Lepoutre, J.M. Roquet, *La personne handicapée mentale et la connaissance du corps humain : un logiciel d'apprentissage*, Mémoire de Licence et Maîtrise en Informatique, Institut d'Informatique, Septembre 1990.
- [Lowe,93] ,Y. Lowette, J-C. Marchal, *Conception d'un générateur de jeux éducatifs*, Mémoire de Licence et Maîtrise en Informatique, Institut d'Informatique, Septembre 1993.
- [Mein,91] ,J.P. Meinadier, *L'interface Utilisateur : Pour une informatique plus conviviale*, Dunod, Paris, France, 1991.
- [Meur,88] ,A De Meur, L. Staes, *Psychomotricité, éducation et rééducation*, De Boeck, Bruxelles, Belgique, 1988.
- [Norm,86] ,D.A.Norman,S.W.Draper, *User Centered System Design : New Perspectives on Human- Computer Interaction*, Lawrence Erlbaum Associates, Londres, 1986.
- [Petz, 92] Charles Petzold, *Programming Windows 3.1 : The Microsoft guide to writing applications for Windows 3.1*. Microsoft Press,Redmond, USA, 1992
- [Rees,94] , Will Rees, Wayne Williams, *Education matter*, in PC HOME, June 1994 pp 83-90
- [Rond,85] ,J-A Rondal,F. Hotyat, *Psychologie de l'enfant et de l'adolescent*, Editions Labor, Bruxelles, Belgique, 1985.
- [Schu,93] ,Ron Schuermans, André Vertens, *Ordinateurs en culottes courtes*, in Computer Magazine Juillet/Aout 93, pp. 100-109.
- [Stro,91] ,Bjarne Stroustrup, *The C++ Programming Language*, Addison-Wesley, 2^e édition, Reading, Massachusetts, USA, 1991.
- [Svan,93] ,D. Svanaes, *Comspec : a software architecture for users with special needs*, Second report, University of Trondheim, Norway, 1993.
- [Vand,93] ,Luc Vandenabeele, *Elaboration d'un logiciel Auteur destiné au domaine du handicap*, Mémoire de Licence et Maîtrise en Informatique, Institut d'Informatique, Septembre 1993.
- [Vand,91] ,J.Vanderdonckt et al., *Une description orientée objet des objets interactifs abstraits utilisé en interface Homme - Machine*, Institut d'informatique, 1991.
-

Annexe : le mode d'emploi d' "Auteur !"

Le mode d'emploi de
"Auteur !" est fourni séparément du
mémoire.

Facultés Universitaires Notre-Dame de la Paix
Institut d'Informatique
rue Grandgagnage, 21, 5000 Namur

Annexe : mode d'emploi du logiciel "Auteur !".

Annexe au mémoire : "Adaptation du système "Auteur !" pour la réalisation d'exercices destinés aux personnes handicapées."

Pascal PLATTEUW

Promoteur : Madame le Professeur Monique NOIRHOMME - FRAITURE

Année académique 1993 - 1994

Mode d'emploi pour le programme "Auteur !"

Auteur : Pascal PLATTEEUW

Table des matières

1. Table des matières

2. Configuration requise	1
---------------------------------	----------

2.1 Configuration matérielle...	1
---------------------------------	---

2.2 Configuration logicielle...	1
---------------------------------	---

2.3 Prérequis...	2
------------------	---

3. Présentation du programme...	2
--	----------

3.1 Qu'est-ce qu'un programme "auteur" ?	2
--	---

3.2 A quoi sert "Auteur !" ?	3
------------------------------	---

3.3 A qui est destiné "Auteur !" ?	3
------------------------------------	---

4. Installation du logiciel	4
------------------------------------	----------

5. Mode d'emploi...	4
----------------------------	----------

5.1 Pour bien démarrer...	4
---------------------------	---

5.2 Les menus...	5
------------------	---

5.2.1 Le menu "FICHIER" ...	5
-----------------------------	---

5.2.2 Le menu "EDITION" ...	9
-----------------------------	---

5.2.3 Le menu "ECRAN"...	10
5.2.4 Le menu "INSERE"...	10
5.2.5 Le menu "INTERACTION"...	11
5.2.6 Le menu "TRACE"...	16
5.2.7 Le menu "EXECUTION"...	16
5.2.8 Le menu "AIDE"...	17
5.3 Le ruban d'outils...	18
5.3.1 Similitudes...	18
5.4 La barre de statut...	18
5.5 Les boîtes de dialogue...	19
5.6 Le mode moniteur vs le mode utilisateur	20
5.7 Les outils...	20
5.7.1 L'icône...	21
5.7.2 L'animation...	23
5.7.3 L'écran...	24

6. Une méthode de construction d'exercices...	34
--	-----------

6.1 Introduction	34
6.2 A vos crayons...	34
6.2.1 Un effort d'imagination...	34
6.2.2 Un plan sur papier...	35
6.3 Passons sur la machine (un exercice mono-écran)...	37
6.4 A vos crayons (un exercice multi-écrans)...	41
6.5 Passons sur la machine (un exercice multi-écrans)...	43
6.6 Méthode par essais et erreurs...	47
6.7 Conclusion	47

7. Annexe A : Ajouter des sons dans la bibliothèque...	48
---	-----------

8. Annexe B : Ajouter des images dans la bibliothèque...	48
---	-----------

9. Annexe C: Troubles du fonctionnement	49
--	-----------

10. Index	50
------------------	-----------

2. Configuration requise...

2.1. Configuration matérielle...

Nous recommandons l'emploi d'un ordinateur de type 386 au minimum. En effet, pour des considérations logicielles que nous verrons plus tard, c'est la machine minimale la plus adaptée pour que ce programme fonctionne bien.

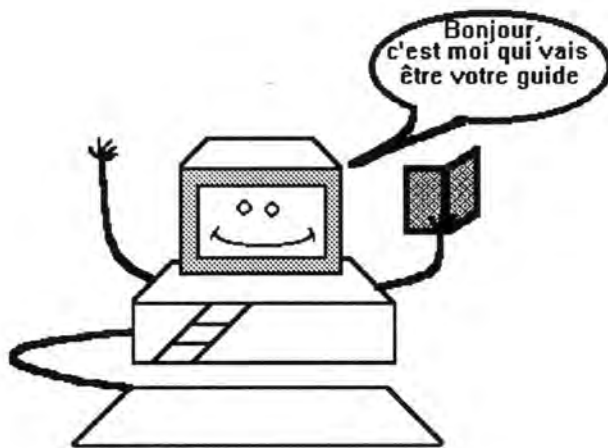
Pour ce qui est de l'écran, le programme a été testé et fonctionne avec les écrans suivants : EGA, VGA, SVGA. Comme le programme démarre en occupant le plus de place possible sur l'écran, la récupération des exercices sauvés ne doit pas poser de gros problèmes.

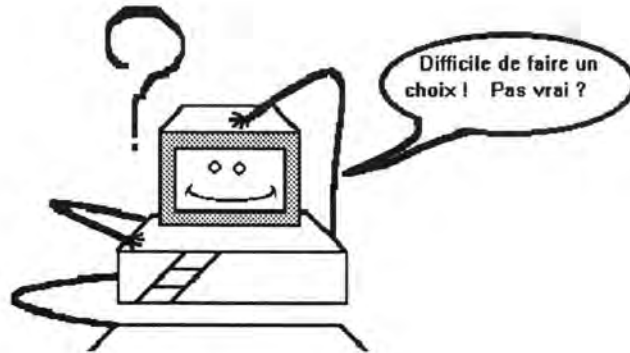
Nous recommandons l'utilisation d'une carte sonore. Toute carte Sound Blaster ou compatible fonctionne avec ce programme pour autant que vous ayez installé les drivers spécifiques à Windows (Cfr. Infra, configuration logicielle). Le non-emploi d'une carte sonore nécessite l'installation d'un ersatz logiciel qui ne rend pas les sons très audibles. Cela est dû entre autres au fait que le haut-parleur de votre PC n'est pas fait pour ce genre de chose.

Le logiciel est prévu pour pouvoir fonctionner à partir du clavier, de la souris, du joystick et d'un contact unique. Toutefois, tout le travail de création d'un exercice se fait à la souris. Celle-ci est donc obligatoire.

2.2. Configuration logicielle...

Le logiciel "Auteur !" ne fonctionnera pas si vous n'avez pas MS-WINDOWS dans sa version 3.X. Ainsi votre configuration mémoire et matérielle est avant tout dictée par Windows.





Les drivers de votre carte sonore doivent être installés avant l'utilisation du logiciel "Auteur !". Si vous n'utilisez pas de carte sonore, les sons que vous entendrez passeront par le haut-parleur de votre PC. Cela nécessite l'emploi d'un driver spécial qui émule une carte sonore. Vous pouvez vous procurer ce driver à l'endroit où vous vous êtes procuré ce programme.

Si vous n'avez pas de carte sonore, il se peut que le programme ne fonctionne pas. Dans ce cas, appelez nous, nous vous dirons comment faire.

2.3. Prérequis...

Ce programme nécessite une certaine habitude de Windows et des termes se rapportant à une interface graphique (notamment les termes "cliquer", "double cliquer",...). Ne vous étonnez donc pas si ces termes apparaissent de temps à autres.

Le deuxième, et le plus important des prérequis demandés par ce programme, est certainement l'habitude de faire certains enchaînements mentaux. Cela veut dire qu'avant de construire un exercice, vous devez savoir ce que vous voulez faire. Le programme ne pourra jamais imaginer un exercice par lui-même ou vous dire que votre exercice n'est pas faisable par un enfant. C'est donc à vous de voir ce que vos utilisateurs peuvent faire et d'imaginer les exercices complets.

3. Présentation du programme...

3.1. Qu'est-ce qu'un programme "auteur" ?

Tout d'abord signalons qu'il n'existe pas un seul type de logiciel auteur. On trouve en effet ce genre de logiciel dans beaucoup de domaines. Citons les principaux : les affaires, la publicité, la simulation, les encyclopédies, les bases de données, les jeux, l'éducation.

Mais qu'est-ce qu'un programme auteur ? Un programme auteur est un programme permettant de regrouper de façon interactive en une seule application des éléments de présentation très différents tels que le son, l'image, la séquence filmée, le texte,... Les interfaces des logiciels auteurs sont très largement imaginées : le fait de cliquer sur un bouton entraîne le déclenchement d'une suite d'action dont l'utilisateur n'a pas à savoir la moindre chose (par exemple : l'exécution d'une macro,...).

Autre caractéristique des logiciels auteur, c'est l'utilisateur qui a le contrôle sur la séquence qu'il crée. Il peut choisir de la modifier à son gré. Ce sont ce qu'on appelle des productions non-linéaires.

3.2. A quoi sert "Auteur !" ?

"Auteur !" est, comme son nom l'indique, un logiciel auteur au sens de la définition ci-dessus. Ce logiciel est destiné au domaine du handicap. Il est destiné à faciliter la tâche des personnes chargées de créer, de construire des exercices pour des enfants handicapés. Sa boîte à outils est, comme nous le verrons par la suite, entièrement graphique. Lorsque l'utilisateur décide de créer une nouvelle icône, il lui suffit de cliquer sur le bon bouton et ensuite de cliquer à l'endroit où il veut voir apparaître cette nouvelle icône. Il peut ainsi insérer, dans les exercices qu'il crée, des images, des sons, des animations, des mouvements, du texte,...

3.3. A qui est destiné "Auteur !" ?

"Auteur !" est un produit qui n'est pas destiné spécialement aux informaticiens. Toutefois, il est bon que les personnes chargées de la création des exercices soient familières avec le maniement des structures algorithmiques de base telles que la séquence, la boucle, la condition.

Une fois l'exercice créé, le logiciel s'adresse à toute personne capable de comprendre cet exercice et de le réaliser. Ce logiciel est suffisamment ouvert pour permettre aux éducateurs et thérapeutes de concevoir des exercices adaptés à la population dont ils s'occupent.

4. Installation du logiciel

Pour installer le logiciel, insérer la première disquette d' "Auteur !" dans le lecteur de disquette, et tapez INSTALL. Suivez les instructions qui apparaissent alors à l'écran.

5. Mode d'emploi...

5.1. Pour bien démarrer...

Avant de démarrer les explications sur le programme proprement dit, il est bon de savoir certaines choses. Ce programme sert aussi bien de bureau de conception que de banc d'essai. Il n'y a pas de programme qui reprenne les exercices créés pour les exécuter. C'est le même programme qui fait le tout. Le présent manuel est conçu de manière à ce que vous puissiez progresser à votre propre rythme. Ainsi, nous présenterons d'abord les éléments de l'interface à laquelle vous allez être confrontés : les menus, la barre d'outils, la barre de status, les boîtes de dialogues,...

Dans un second temps, nous vous dirons de quels outils vous disposez et quelles actions vous pouvez leur appliquer. Puis nous vous présenterons les concepts de base du programme : l'icône, l'écran, l'exercice, la séance.

Enfin, nous vous apprendrons à personnaliser les exercices en fonction des personnes qui auront à les faire, et nous tâcherons de vous expliquer comment garder une trace de ce que l'élève a fait.

Vous êtes prêt ? Alors cliquez deux fois sur l'icône de "Auteur !" et en avant pour l'explication du programme.

5.2. Les menus...

5.2.1. Le menu "FICHIER"...

Le menu "**FICHIER**" est un menu très important pour tout créateur d'exercices. C'est en effet à partir de ce menu que se commandent les actions de sauvegarde des exercices créés. Nous allons passer en revue les différentes commandes proposées dans ce menu.

La première de ces commandes est la commande "**Nouveau**".

Elle sert à commencer un nouvel exercice. C'est dans ce mode que le programme commence. C'est-à-dire que lorsque vous lancez le programme, celui-ci commence un nouvel exercice sans que vous ayez à faire quoi que ce soit.

En fait cette commande du menu "Fichier" sert lorsque vous avez construit un exercice. Vous venez de sauvegarder l'exercice en cours et vous voulez en créer un autre, qui n'a rien à voir avec celui que vous venez de faire.

Le problème, c'est que l'ancien exercice est encore à l'écran... Donc il faut signaler au programme que vous voulez commencer quelque chose de "nouveau". Mais attention, le programme ne vous demandera pas de confirmation dans le cas où vous n'auriez pas sauvé votre exercice. La seule question qui vous sera posée est "Etes-vous sûr de vouloir faire un nouvel exercice ? O/N". Si vous répondez "OUI" et que vous n'avez pas sauvé votre exercice préalablement, ce dernier est perdu.

Cette commande du menu est accessible par le clavier en frappant simultanément les touches "**ALT - F**" puis en poussant sur "**N**".

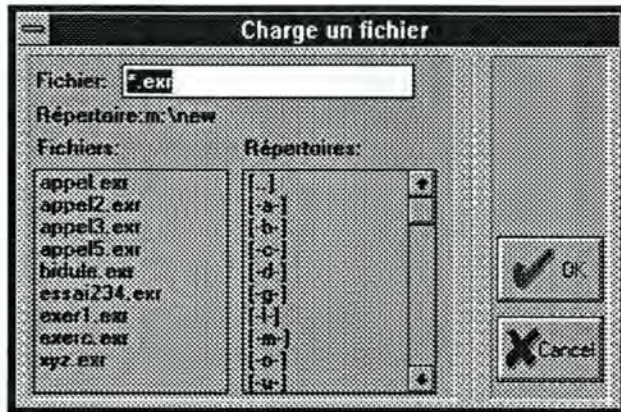
N'oubliez jamais de sauver très régulièrement (Cfr. ci-dessous) ce que vous êtes en train de faire. Un accident, une maladresse, voire une panne est si vite arrivé.

24/07/94	
Fichier	Edition Ecran Insère
Nouveau	
Ouvrir...	F3
Ouvrir écran	
Enregistrer écran...	
Enregistrer tout	F2
Préférences...	
Quitter	Alt+x



La seconde commande proposée est la commande "**Ouvrir**".

"Ouvrir" est un mot informatique qui désigne généralement l'action d'aller chercher sur le disque ou sur la disquette le contenu d'un fichier. Dans ce programme vous ne pouvez "ouvrir" que des exercices que vous avez préalablement sauves. Si c'est la première fois que vous vous servez de "Auteur !", il se peut qu'il n'y ait pas d'exercices sauves sur le disque ou la disquette. Le programme ne vous laissera donc pas ouvrir un fichier puisqu'il n'y en a aucun existant.



S'il y a des exercices sauves sur le disque ou sur la disquette, le programme vous les propose dans une liste. Si l'exercice que vous voulez ouvrir ne se trouve pas dans cette liste, il se peut qu'il se trouve dans un autre répertoire du disque. Il vous suffit d'explorer celui-ci répertoire par répertoire ou de taper le nom du répertoire et celui du fichier in extenso dans la case prévue à cet effet. Ci-contre, vous trouverez une représentation de la boîte de dialogue à laquelle vous serez confrontés quand vous voudrez ouvrir un fichier.

La commande "**Ouvrir**" est accessible à partir du clavier en frappant d'abord les touches "**ALT + F**" puis en poussant sur "**O**" ou encore en poussant sur la touche **F3**.

Suivant la commande "**Ouvrir**", vient la commande "**Ouvrir écran**". Cette commande sert à recharger les écrans que vous avez sauves. Cette commande ne ressemble pas à la précédente : elle n'ouvre qu'un seul écran. La commande précédente ouvrait tout un exercice. Autre différence , "Ouvrir" ouvrait un nouvel exercice en effaçant éventuellement l'exercice en cours. "Ouvrir écran" n'efface pas l'exercice en cours. Cette action ajoute l'écran que vous ouvrez à la suite de ceux que vous avez déjà créés.

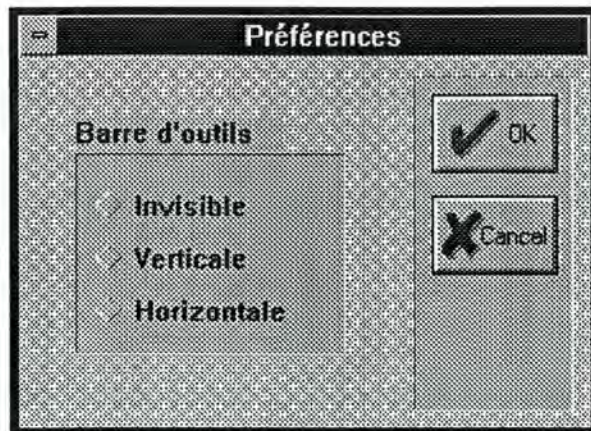
La quatrième commande de ce menu est certainement la plus utilisée sur n'importe quel programme. C'est en effet cette commande qui permet de sauver les exercices. Il s'agit de la commande "**Enregistrer**".

"Enregistrer" est un terme qui signifie que vous voulez garder l'exercice que vous venez de réaliser sur une unité disque (que ce soit un disque dur ou une disquette). La boîte de dialogue est presque la même que pour charger un fichier en mémoire. Donc si vous ne voyez pas le fichier d'exercice que vous cherchez, pas de panique. Ce fichier est peut-être dans un autre répertoire, ou sur une autre unité de disque. Un peu de recherche dans les répertoires ou sur les disques vous permettra de retrouver le fichier exercice en question. Il vous suffit alors de le sélectionner et puis d'appuyer sur le bouton **"OK"**. Si vous estimez que l'exercice ne vaut pas la peine d'être sauvé, ou si vous arrivez dans cette boîte de dialogue à la suite d'une fausse manoeuvre, vous pouvez toujours cliquer sur le bouton **"Annuler"** ou **"Cancel"**

Vous remarquerez qu'il y a deux (2) façon de sauver des choses que vous avez faites. Vous pouvez **"Enregistrer"** un écran, **"Enregistrer"** tout l'exercice. **"Enregistrer"** un écran est une possibilité laissée à l'utilisateur de sauver un écran dont il est particulièrement satisfait ou un écran qui peut resservir dans un autre exercice. **"Enregistrer"** tout l'exercice ne vous laisse pas le choix d'enregistrer un élément particulier. Vous donnez un nom à l'exercice dans son entièreté.

Il y a d'autres façons d'atteindre ces fonctions dans le menu :

- **"Enregistrer écran"** peut s'atteindre en poussant les touches suivantes : **ALT + F** puis **E**.
- **"Enregistrer tout"** peut être atteint par : **ALT + F** puis **T** ou par **F2**.



La commande suivante, "**Préférences**" sert à effectuer différents réglages. Et ce afin que vous puissiez personnaliser votre copie du logiciel. Lorsque vous sélectionnez cette commande, la boîte de dialogue ci-contre apparaît.

Elle vous permet de modifier la position de la barre d'outils. Si vous cochez la case "verticale", la barre apparaîtra verticalement, à gauche de l'écran. Si vous cochez la case "horizontale", la barre apparaîtra horizontalement au dessus de l'écran. Si vous cochez la case "invisible", la barre d'outils disparaîtra de l'écran. Il vous faudra décocher la case (enlever la marque dans la case marquée "invisible") pour qu'elle réapparaisse.

Cette boîte de dialogue peut aussi être obtenue en appuyant sur les touches : **ALT +F** puis **P**.

La dernière commande de ce menu est la commande "**Quitter**". Nous pensons qu'il n'y a pas grand chose à écrire sur cette commande. Signalons simplement qu'elle peut être atteinte par les touches **ALT +F** et **Q** ou **ALT** et **X**.

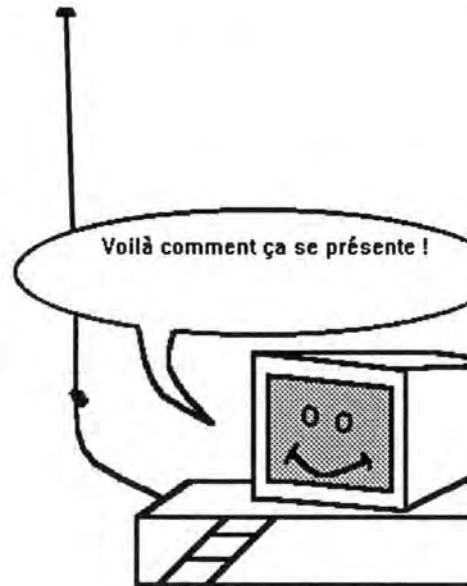
5.2.2. Le menu "EDITION"...

Le menu "EDITION" contrairement aux autres applications Windows, ne comporte pas les possibilités de ce que l'on appelle le "Couper-Coller". Nous avons estimé que ces instructions ne servaient pas dans un tel programme. Ce menu ne comporte donc que deux instructions.

La première de ces instructions est la commande "Effacer". Elle vous permet de supprimer définitivement un élément de l'exercice. Cet élément doit être préalablement sélectionné au moyen de l'outil en forme de flèche. Un élément sélectionné se remarque grâce aux quatre petits rectangles blancs figurant aux quatre coins.

Son raccourci clavier est simple : soit **ALT + E** puis **E**, soit la touche **DEL**.

La deuxième et dernière commande est assez simple à comprendre. Elle sert simplement à sélectionner tous les éléments d'un écran. C'est la commande "**Sélectionner tout**" et son raccourci est **ALT + E** puis **N**. On ne pouvait pas faire plus simple. "**Sélectionner tout**" peut s'avérer pratique lorsque l'on veut tout effacer de l'écran (si par exemple, on s'aperçoit que ce qu'on a fait jusqu'à présent n'est pas bon du tout).



5.2.3. Le menu "ECRAN" ...

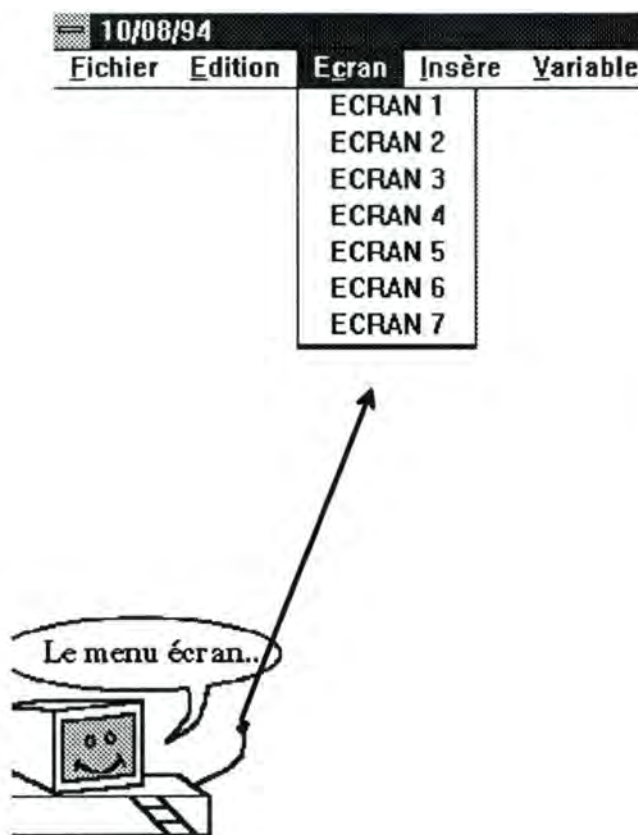
Le menu "ECRAN" est un menu un peu spécial : C'est le seul menu variable du programme. Sa longueur varie en effet selon le nombre d'écrans que vous créez. Quand vous démarrez le programme ou que vous choisissez de créer un nouvel exercice, ce menu indique "ECRAN 1". Chaque fois que vous insérez un écran en poussant sur le bouton prévu à cet effet ou en choisissant de l'insérer à partir des menus, un écran "ECRAN X" (où X est le numéro de l'écran) apparaît dans ce menu. Pour passer d'un écran à l'autre, il vous suffit de choisir votre écran de destination dans ce menu. Celui-ci sera alors amené au premier plan et cachera les autres écrans.

Notez que l'écran actif - c'est-à-dire celui dans lequel vous vous trouvez - est signalé dans ce menu par un symbole ("3"). Il n'y a aucun raccourci clavier si ce n'est le raccourci pour accéder au menu : **ALT + C** après, vous devez naviguer dans le menu à l'aide des flèches de direction de votre clavier ou cliquer sur l'item que vous désirez voir apparaître.

5.2.4. Le menu "INSERE"...

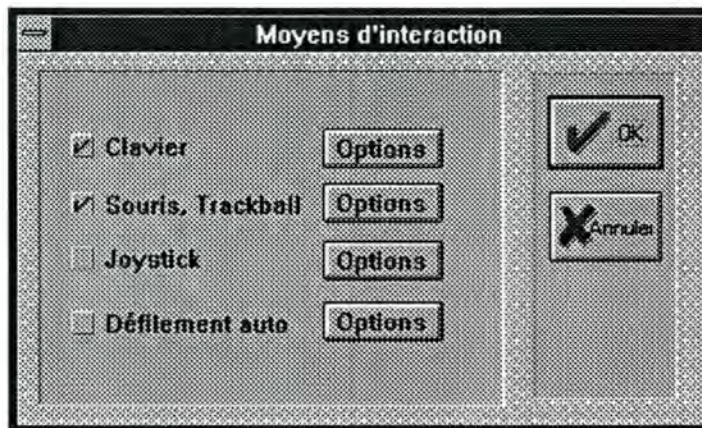
Le menu "INSERE" est en fait une répétition de la barre d'outils que vous voyez sur le côté gauche ou en haut de votre écran. Il existe simplement à cause du fait que l'on peut rendre la barre d'outils invisible. Or si celle-ci est invisible, vous n'avez plus accès à ses éléments. Grâce à ce menu, vous pouvez toujours travailler avec les outils même s'ils n'apparaissent plus à l'écran. Nous approfondirons les différentes fonctions de ce menu quand nous aborderons les outils que vous avez à votre disposition (Cfr. 5.3).

Remarquez que lorsque vous sélectionnez une fonction de ce menu, le programme réagit comme si vous aviez appuyé sur le bouton de la barre d'outils. La seule action visible à vos yeux est le changement de forme du curseur. Le reste vous appartient.



5.2.5. Le menu "INTERACTION"...

Le menu "INTERACTION" est un des plus intéressants menus qui soit dans ce programme. Il permet en effet de régler le ou les moyens d'accès au programme. Ainsi, si les personnes auxquelles vous destinez les exercices ne savent pas se servir de la souris ou du clavier, il n'y a aucun problème. Le programme est prévu pour fonctionner avec un joystick ou un contact unique (un switch). Et ce dans sa partie exécution d'exercice.



Quand vous cliquez sur le menu "INTERACTION", il n'y pas de sous-menu qui apparaît. A la place, surgit une boîte de dialogue. Celle-ci (présentée ci-contre) permet le choix du moyen d'accès en fonction des possibilités des gens qui devront pratiquer les exercices que vous créez.

Vous avez le choix. Mais, pour des raisons pratiques, il nous est apparu nécessaire de faire savoir au programme que par défaut, le clavier et la souris sont actifs. C'est pourquoi les cases de la souris et du clavier sont cochées.

A côté de chaque moyen d'interaction se trouve un bouton "Option". Ce bouton permet dans tous les cas, d'effectuer un réglage plus fin du moyen d'interaction en face duquel il se trouve.

Ainsi, pour le clavier, en poussant sur le bouton "Option", on a accès à une nouvelle boîte de dialogue. Au milieu de celle-ci se trouve une barre de défilement. Cette dernière permet le réglage du délai de répétition d'une touche. C'est-à-dire que si vous augmentez cette valeur, il faudra pousser plus longtemps sur la touche pour que celle-ci se répète et inversement.

Cette boîte de dialogue se présente de la manière suivante :

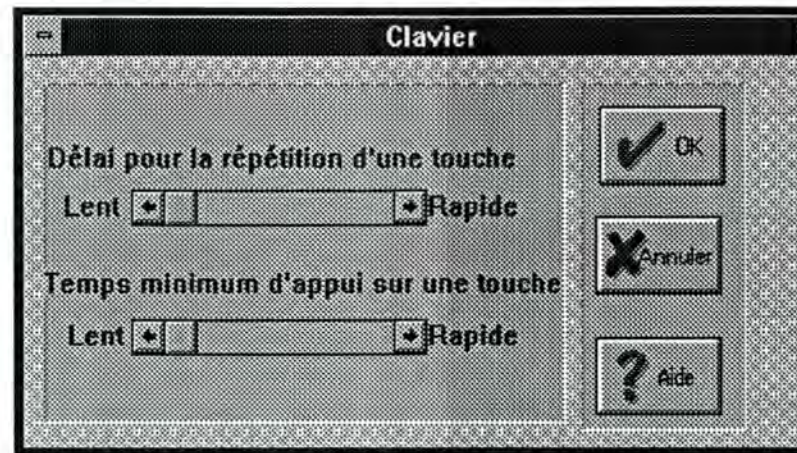


Figure 1 : Gestion des paramètres du clavier.

Remarquez qu'une aide vous est proposée si vous ne comprenez pas ce qu'il faut faire. Pour y accéder, cliquez simplement sur le bouton avec le point d'interrogation. Une nouvelle fenêtre apparaîtra alors pour vous renseigner sur ce qu'il y a lieu de faire.

La souris se règle par une boîte de dialogue similaire. Notez que la vitesse de la souris se règle indépendamment verticalement et horizontalement. Vous pouvez aussi régler la vitesse du double click sur le bouton gauche de la souris. Nous permettons ce réglage bien que nous vous déconseillons de faire réagir les exercices que vous créez au double click. Le double click est un concept difficile à maîtriser, même pour une personne valide.

La boîte de dialogue ressemble à ceci :

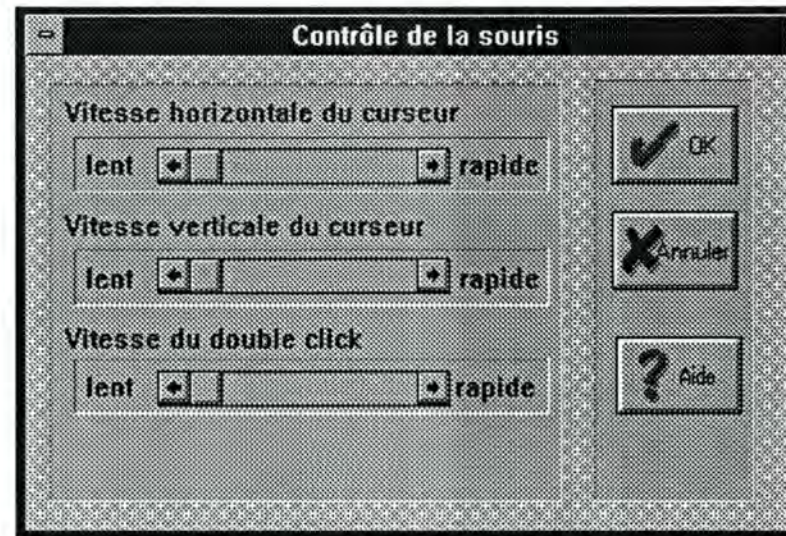
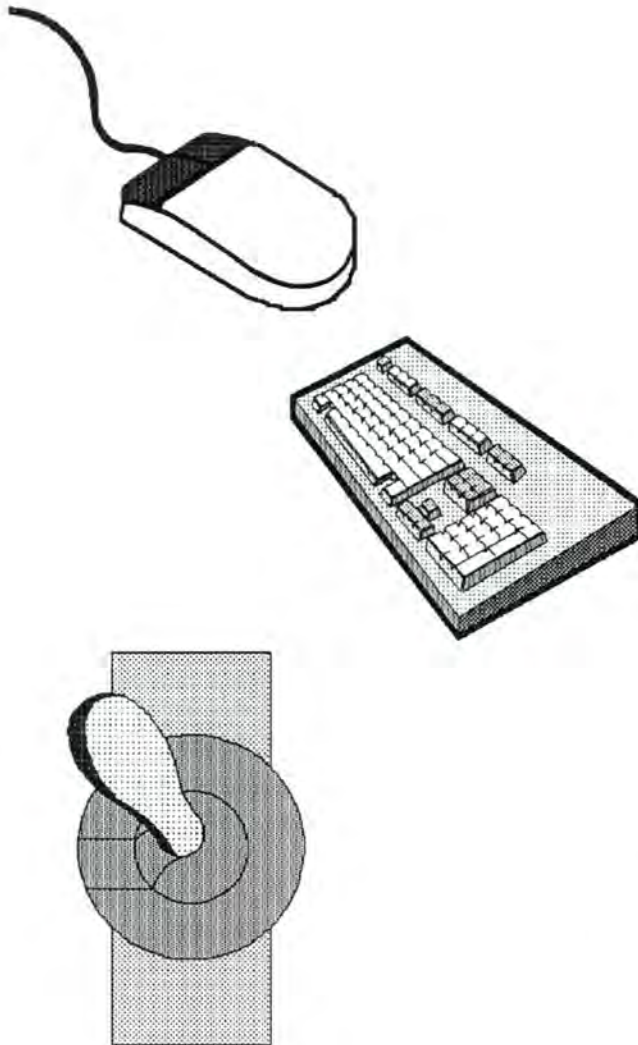


Figure 2 : Gestion des paramètres de la souris.

Ici encore, une aide en ligne vous permet de savoir quoi faire au cas où vous hésitez. Là encore, vous y accédez par le bouton marqué d'un point d'interrogation.

Le joystick peut être, lui aussi, réglé par une boîte de dialogue. Cette dernière permet de régler la vitesse de déplacement du curseur et la forme du curseur. Le joystick sert principalement dans les exercices de navigation. Il est particulièrement bien prévu pour ce genre d'exercice.

Un bouton permet de calibrer le joystick. Calibrer le joystick, c'est dire à l'ordinateur quelle amplitude on peut donner au mouvement du joystick. C'est le calibrage qui permet de dire où se trouve la limite gauche-droite et la limite haut-bas.

Sa boîte de dialogue est la suivante :

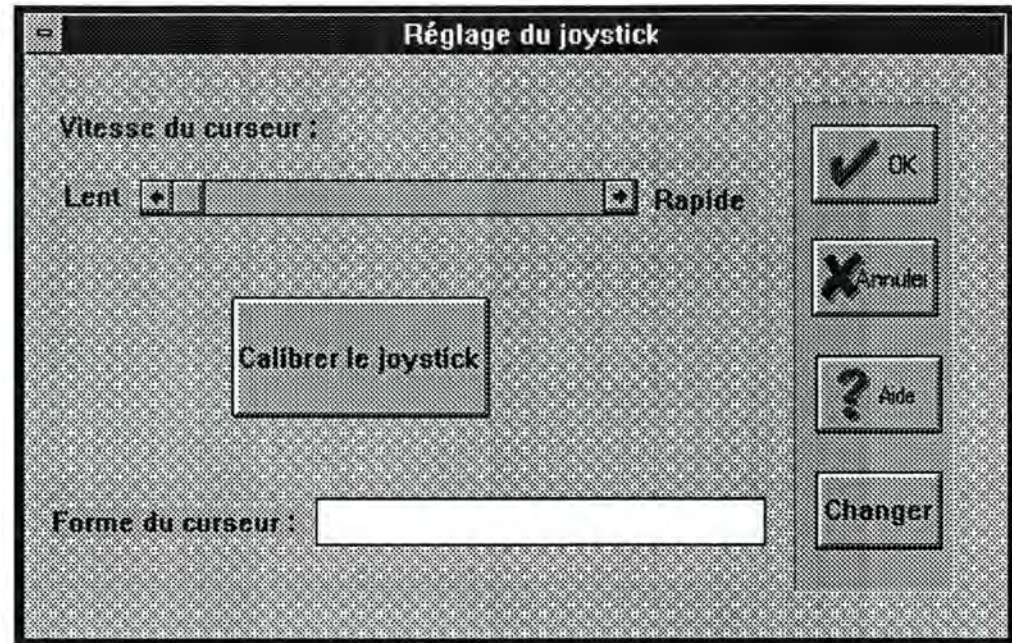


Figure 3 : Gestion des paramètres du joystick.

Le bouton "Changer" permet l'accès à une boîte de dialogue de chargement de fichiers. Les fichiers qui sont proposés sont des fichiers contenant des formes de curseur. Le bouton marqué d'un point d'interrogation est, comme d'habitude, réservé à l'aide en ligne du programme.

Le dernier moyen d'interaction est le "switch", également appelé "contact unique". Il permet de manipuler la partie utilisateur du programme grâce au seul bouton gauche de la souris. Les déplacements de la souris n'ont plus d'importance : ils ne sont pas pris en compte quand ce moyen d'interaction est sélectionné.

La boîte de dialogue permettant de régler le "switch" est la suivante :

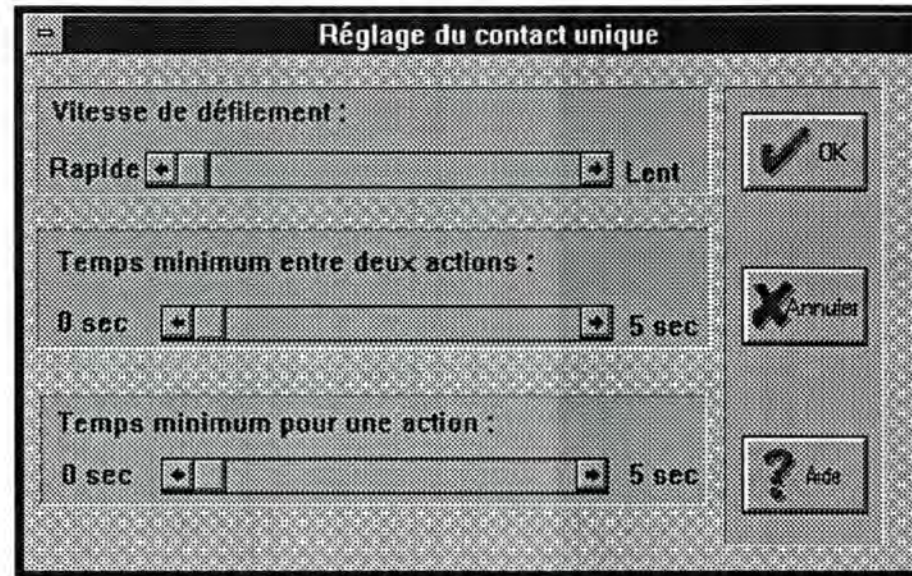


Figure 4 : Gestion des paramètres du switch.

5.2.6. Le menu "TRACE"...

Le menu "**TRACE**" est un menu très intéressant. C'est de ce point du programme que vous pourrez suivre l'évolution des utilisateurs dans la pratique des exercices. Encore une fois, lorsque vous sélectionnez cette option du menu, il n'y a pas de sous-menu.



A la place, une boîte de dialogue apparaît. Dans cette boîte de dialogue, vous pouvez donner le nom de la personne que vous suivez ainsi que son âge. La trace est un mécanisme très simple. Lorsque vous déclenchez la trace (en cochant la case prévue à cet effet dans la boîte de dialogue) et que vous passez en mode utilisation, tous les clics de l'utilisateur dans l'écran sont enregistrés dans le fichier. On enregistre ainsi le moment où l'utilisateur a cliqué et l'endroit où il a cliqué.

Tous ces renseignements sont enregistrés dans un fichier texte (par défaut le fichier TRACE .TRA) que vous pouvez consulter à partir de n'importe quel éditeur de texte.

La trace n'est pas déclenchée automatiquement. Il faut que vous la demandiez explicitement.

5.2.7. Le menu "EXECUTION"...

Le menu "**EXECUTION**" est le menu le plus important de ce logiciel. C'est à partir de celui-ci que les exercices que vous créez s'exécutent. Lorsque vous sélectionnez cet item du menu, vous passez dans ce que nous appelons le mode Utilisateur (Cf. infra 5.6 Le mode moniteur vs le mode utilisateur).

A partir de là, ce n'est normalement plus vous qui êtes devant le clavier et l'écran, mais la personne pour qui vous avez créé l'exercice. Toutefois, il se peut que ce soit encore vous, alors ne partez pas tout de suite. Le mode "Utilisateur", hormis qu'il soit destiné à un autre type d'utilisation, peut être très pratique pour tester un exercice que vous venez de finir.

On passe en mode "Utilisateur" en sélectionnant l'item **"EXECUTION"** dans le menu ou en appuyant simultanément sur les touches **ALT** et **U**. Pour revenir dans le mode "Moniteur", il vous faudra interrompre l'exercice en cours. Mais ici, il n'y a plus de menu (la barre de menu a été supprimée pour que l'attention de l'utilisateur ne soit attirée par rien d'autre que l'exercice en cours). D'où la possibilité d'interrompre l'exercice en cours par une combinaison de touches.

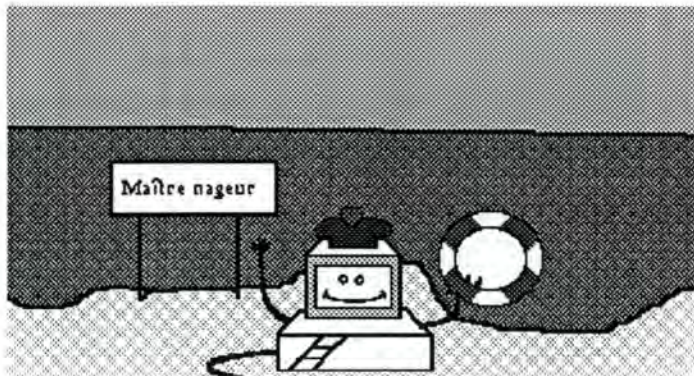
Cette combinaison est **ALT** et **M**. Ainsi vous revenez au mode "Moniteur" et vous pouvez éventuellement corriger l'exercice afin de l'adapter à la personne.

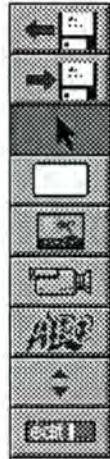
5.2.8. Le menu "AIDE"...

Comme son nom l'indique, ce menu vous fournit (normalement) une aide sur le programme. En cliquant sur les phrases vertes, vous passez d'information en information. A vous de définir le chemin d'accès vers un sujet à propos duquel vous voulez plus de renseignements.

Cette aide peut être obtenue par appui sur la touche **F1**.

Il y a un autre item dans ce menu : c'est l'item **"A PROPOS DE"**. Cet item n'apporte rien au programme sinon des renseignements sur la version du logiciel, sur ses auteurs, sur l'endroit où on peut se le procurer et éventuellement se procurer des renseignements sur des problèmes non signalés. Surtout n'hésitez jamais à vous servir du téléphone inscrit dans cette boîte de dialogue. Mais avant de téléphoner, notez le problème sur une feuille de papier afin d'être le plus précis et le plus concis possible. Nous nous ferons un plaisir de vous aider en toutes circonstances.





5.3. Le ruban d'outils...

Le ruban d'outils se présente comme une série de boutons poussoirs qui sont à votre disposition pour faciliter la réalisation des exercices. Ces boutons représentent tous un outil : une icône, une animation, un écran, un défilement, un texte,...

L'appui sur un de ces boutons change la forme du curseur. Celui-ci prend la forme de l'outil sélectionné. Un click sur le bouton gauche de la souris a, alors, pour effet de placer, à l'endroit du click, l'objet désiré.

Si vous cliquez dans le ruban sur l'outil "Icône" lorsque vous cliquez dans l'écran, vous définissez le coin supérieur gauche de l'icône dessinée. C'est la même chose lorsque vous définissez les autres outils. Seul le click sur l'outil "écran" ne change pas le curseur. Quand vous choisissez cet outil, un nouvel écran est inséré dans l'exercice.

5.3.1. Similitudes...

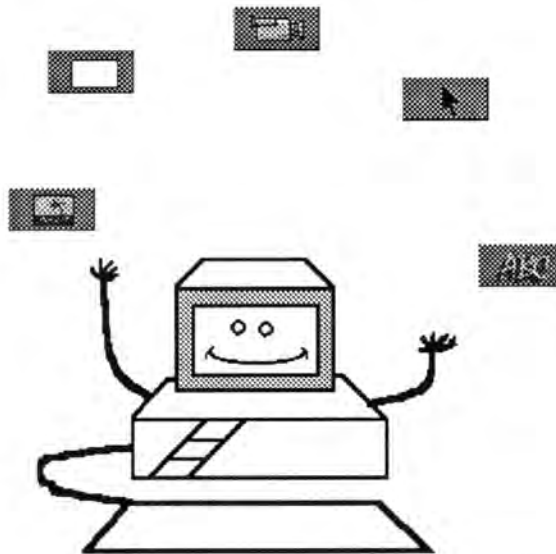
Le ruban est en fait l'outil dont vous vous servirez sans doute le plus. Toutefois, nous l'avons dit ci-dessus, il se peut que vous décidiez de le rendre invisible pour pouvoir travailler sur un écran plus grand. Dans ce cas, le ruban peut être simulé par le menu "INSERE". Une sélection dans ce menu revient au même qu'un click sur un des outils du ruban.

5.4. La barre de statut...

La barre de statut se trouve en bas de l'écran. Elle n'apparaît qu'en mode moniteur. Elle vous renseigne sur les actions que vous êtes en train de faire. C'est dans cette zone qu'apparaissent normalement certains messages d'aide, ou des renseignements sur l'action qui suit celle que vous venez de poser.

5.5. Les boîtes de dialogue...

Le programme repose en fait beaucoup sur le remplissage de boîte de dialogue. C'est par leur intermédiaire que vous pourrez donner certains attributs à vos icônes et animations. Toutefois, il y a certains termes qu'il vous faut dès maintenant connaître sous peine de ne pas comprendre quels attributs vous donnez à vos créations.



Dans chaque boîte de dialogue figurent normalement trois boutons. Ces trois boutons sont :

- ♦ OK,
- ♦ ANNULER,
- ♦ AIDE.

Le bouton OK permet de dire au programme : "J'ai terminé mes réglages. Tu peux les appliquer.". Le bouton ANNULER permet de revenir à l'écran précédent sans prendre en compte les modifications survenues. Le bouton AIDE permet d'obtenir de l'aide sur l'action que l'on entreprend.

En dehors de ces trois boutons, il se peut que vous en rencontriez d'autres : AJOUTER, SUPPRIMER, CHARGER, CHANGER,... Ces boutons permettent généralement de changer l'image d'une icône, d'ajouter ou de supprimer des images dans une animation,...

Certaines boîtes de dialogue sont composées de barres de défilement. Celles-ci permettent de changer certains paramètres du système. Utilisez-les avec beaucoup de précaution. Bien qu'aucun changement ne soit définitif, il vous faudra peut-être sortir de Windows pour retrouver une situation normale. Si cela devait se produire trop souvent, contactez-nous, nous essaierons de régler le problème.

Ces boîtes de dialogue sont généralement assez claires pour que vous n'ayez pas de problèmes. Une explication plus détaillée vous sera fournie plus loin dans le manuel sur certains détails que vous devez savoir.

5.6. Le mode moniteur vs le mode utilisateur

Comme vous le savez maintenant, il y a deux modes de fonctionnement du programme. Il y a d'une part le mode "Moniteur", dans lequel vous créez et vous modifiez des exercices, et d'autre part, le mode "Utilisateur", dans lequel vous pratiquez ou essayez les exercices réalisés dans l'autre mode.

On passe du mode "Moniteur" au mode "Utilisateur" en cliquant dans le menu sur l'item **Exécution** ou en appuyant simultanément sur les touches **ALT** et **U**. Pour l'opération inverse, il n'y a plus de menu présent à l'écran. La seule façon de faire pour repasser en mode "Moniteur", est d'appuyer, toujours simultanément, sur les touches **ALT** et **M**.

Dans le mode "Moniteur", vous pouvez utiliser tous les outils, les menus, ... Dans le mode "Utilisateur", vous n'avez plus accès qu'aux éléments dits "sélectionnables" et par le(s) moyen(s) d'accès défini en paramètre.

Ainsi, si le mode d'interaction choisi est le clavier et la souris, vous pouvez vous servir du clavier et de la souris dont vous voyez le curseur. Si le mode d'accès est le joystick, seules les combinaisons de touches ci-dessus sont effectives à l'exclusion de toute autre. Enfin, si le mode d'accès est le contact unique, le curseur de la souris n'apparaît plus et vous devez nécessairement disposer d'un contact unique.

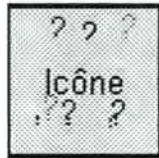
5.7. Les outils...

Comme nous vous l'avons dit lorsque nous avons parlé du ruban d'outils, il y a à votre disposition, une série d'outils à partir desquels vous pouvez construire des exercices. Ces outils vous sont présentés plus en détails ci-après. Nous allons vous expliquer quels paramètres on peut changer et ce que ça signifie au niveau du programme.

5.7.1. L'icône...

L'icône est le premier outil que nous vous présentons. C'est l'un des principaux outils de ce programme. Bien des exercices peuvent déjà être construits avec lui seul.

Sur le ruban, il se trouve en cinquième position.



Pour placer une icône à l'écran, cliquez sur le bouton de l'icône ou choisissez l'item "**Icône**" dans le menu "**Insère**". Lorsque c'est fait, revenez en mode édition en cliquant, dans le ruban, sur la flèche.

Vous voilà prêt. Cliquez deux fois (autrement dit "double-cliquez") sur l'icône que vous venez de poser. Une boîte de dialogue apparaît. Cette boîte est intitulée **ICONE**. Vous remarquerez qu'il y a toute une série de paramètres réglables à partir de cette boîte. Passons-les en revue.

Le premier paramètre est le nom de l'icône. Par défaut, ce nom est "Icône X" où X désigne le numéro de l'icône. Ce numéro dépend évidemment du nombre d'icônes que vous avez placées sur l'écran. Vous remarquerez aussi que ce nom est en surbrillance. De ce fait, si vous tapez quelque chose au clavier, ce que vous tapez remplace le nom par défaut de l'icône. Je vous propose de remplacer le nom par défaut (normalement "Icône 1") par un nom que vous pouvez choisir. Voilà votre première icône personnalisée.

Le deuxième groupe de paramètres concerne la position et la taille de l'icône. Il va de soi qu'il est plus facile de modifier ces paramètres en déplaçant l'icône sur l'écran qu'en les modifiant à la main dans la boîte de dialogue. Nous vous expliquerons plus tard comment procéder à ces modifications.

Vient ensuite un groupe concernant les propriétés de l'icône :

- La première de ces propriétés est "Visible". Cela signifie que l'icône sera toujours visible.
- La seconde est "Sélectionnable". Cela signifie que si la case correspondant à cette propriété est cochée, lorsque vous passerez en mode "Utilisateur", ce dernier pourra, au moyen de son moyen d'interaction, sélectionner l'icône. Si vous souhaitez placer une icône non sélectionnable, il vous suffit de décocher cette case.
- La troisième est "Amovible". Elle signifie que lorsqu'un utilisateur cliquera dessus, elle disparaîtra à ses yeux.
- La dernière est "Pile". Elle signifie que lorsqu'un utilisateur cliquera dessus, il créera une copie de cette icône. qu'il déplacera sur l'écran pour, par exemple, la poser en un endroit précis.

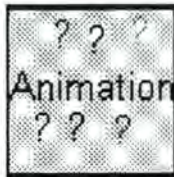
A droite de ce groupe de propriétés, il y a encore une case. Si elle est cochée, elle fait apparaître une bordure autour de l'icône. Sinon l'icône apparaît sans bordure.

Enfin, tout en dessous, se trouve un champ contenant le nom du fichier qui est chargé par défaut pour illustrer l'icône. Ce fichier s'appelle IM.BMP. Ne le déplacez jamais du répertoire où il est, cela provoquerait sans doute des problèmes dans le programme. Vous pouvez bien sûr changer ce fichier par défaut. Il vous suffit pour cela de cliquer sur le bouton OUVRIER. Une nouvelle boîte de dialogue apparaît alors. Elle vous permet de choisir un nouveau fichier. Remarquez que tous ces fichiers portent la même extension. Le nouveau fichier chargé remplace l'illustration par défaut.

Dans la boîte de dialogue, vous aurez sûrement remarqué qu'il y a un bouton ACTIONS. Nous vous reparlerons plus tard de ce bouton et de ce que l'on peut faire à partir de celui-ci. Mais ne nous en préoccupons pas maintenant.

5.7.2. L'animation...

Le second outil que nous vous proposons d'étudier se trouve en sixième position sur le ruban. Il est représenté par une petite caméra.



L'animation est en fait une suite d'images (10 au maximum) qui s'enchaînent les unes après les autres. Il ne s'agit donc pas d'une image qui bouge dans l'écran. L'animation est donc essentiellement statique. Elle convient très bien pour des renforcements positifs et/ou négatifs.

Cliquez, dans le ruban, sur la caméra. Placez votre animation sur l'écran et revenez en mode "Edition". Cliquez deux fois sur l'animation ainsi placée.

La boîte de dialogue qui apparaît vous permet de changer les attributs de l'animation. Le premier que vous pouvez changer est le nom de l'animation. Le nom par défaut est "Animation X" où X correspond à un chiffre indiquant le numéro de l'animation. Il ne tient qu'à vous de donner un nom plus explicite à vos animations.

Ensuite, vous pouvez changer la position de l'animation. Toutefois, pour faire cela, nous vous conseillons d'utiliser la souris. Quelques clics appropriés doivent suffire. Vous pouvez changer cette position mais cela implique, et ce n'est pas facile, que vous calculiez les nouvelles coordonnées écran de votre animation. Vous pouvez aussi changer la taille de l'animation. Mais, même remarque, il est plus facile de le faire à l'aide de la souris. Ces quatre champs de taille et de position ne sont là qu'à des fins d'alignement. Si vous décidez qu'il faut que vos icônes ou vos animations soient alignées, il peut être plus facile de les aligner sur la même coordonnée. C'est à cette seule fin que ces champs existent.

Après les champs de position et de taille, vient une case à cocher portant la mention "sélectionnable". Cela veut dire que vous pouvez associer des actions à une animation et que si l'utilisateur sélectionne l'animation, ces actions s'exécuteront selon le même principe que pour les icônes. Si vous ne souhaitez pas que l'animation soit sélectionnable, il vous suffit de décocher cette case.

Enfin, les derniers paramètres que vous pouvez changer sont les images faisant partie de l'animation. Vous pouvez déclarer une suite de dix (10) images. Ces dix images constitueront l'animation. Elles seront enchaînées lors de l'exécution de l'animation. Pour ajouter une image dans la suite, il vous suffit de cliquer sur le bouton "Ajoute..." et de choisir parmi les images proposées. Pour supprimer une image de la liste, il vous faut sélectionner l'image et ensuite seulement appuyer sur le bouton "Supprime".

5.7.3. L'écran...

L'écran est le concept clef de ce programme. C'est en effet sur un écran que vous allez définir vos exercices. Au démarrage, un écran est déjà créé. Son nom par défaut est "ECRAN 1". Nous allons voir, dans un premier temps, comment ajouter un nouvel écran, et, dans un deuxième temps, comment changer les paramètres d'un écran.

Pour ajouter un écran, il vous faut sélectionner l'outil "écran" dans la barre d'outils. Cet outil n'a pas d'effet visible. Tout au plus, vous aurez l'impression que le programme a effacé tout votre travail. Rassurez-vous, il n'en est rien. Le programme a juste créé un nouvel écran au dessus de l'écran en cours. Vous remarquerez aussi que, contrairement aux autres outils, le curseur de l'outil de création d'un écran ne change pas du curseur courant.

Pour contrôler que tout s'est bien passé, il faut aller voir le menu "Ecran". Ce menu contient la liste des écrans déjà créés. C'est un menu à longueur variable : plus vous créez d'écrans, plus il est long. Si tout s'est bien passé, il doit y avoir un nom de plus dans ce menu.

Voyons maintenant comment modifier les paramètres d'un écran. Pour faire cela, double-cliquez sur le fond de l'écran. Une boîte de dialogue apparaît. Cette boîte contient quatre boutons. Les deux premiers servent à accepter ou à refuser les changements que vous apportez à l'écran. Le troisième sert à associer des actions à l'écran. Ainsi, lorsque vous parviendrez à cet écran au cours d'un exercice, les actions associées à l'écran s'exécuteront d'elles-mêmes, sans intervention de l'utilisateur. La boîte de dialogue qui apparaît à ce moment est la boîte de dialogue classique d'édition des actions.

Le quatrième bouton présent dans cette boîte de dialogue permet le changement de la couleur de fond de l'écran. Un click sur ce bouton fait apparaître une boîte de dialogue classique d'édition de couleurs. Vous pouvez choisir la couleur du fond de votre écran. Et si d'aventure les couleurs proposées ne vous conviennent pas, vous pouvez toujours définir des couleurs personnalisées.

5.7.4. Le texte...

Le texte est un élément qui vous permet de mettre des commentaires dans un exercice ou de placer des explications sur le but de l'exercice (en supposant que les personnes à qui vous le destiner sachent lire). Pour placer du texte à l'écran, sélectionner l'outil "Texte" dans la barre d'outils. Cet outil est représenté par les trois lettres "ABC" en bleu sur fond gris. Une fois que le curseur a changé, il vous suffit de cliquer à l'endroit où vous voulez voir apparaître votre texte. A l'endroit où vous cliquez, un texte par défaut apparaît. Pour changer ce texte, cliquez dessus deux fois avec l'outil d'édition (la flèche). Une boîte de dialogue apparaît. Celle-ci vous permet de changer la taille des caractères, la forme des caractères (en changeant la police), la couleur des caractères,...

Le texte n'est normalement pas destiné à être sélectionné. Il est juste destiné à informer les personnes qui utilisent "Auteur!", que ce soit les créateurs d'exercices ou les utilisateurs.

5.7.5. La ligne...

La ligne est le dernier des outils que nous vous présentons dans ce manuel d'utilisation. La ligne se trace directement sur l'écran. Il n'y a pas de fenêtre associée. L'outil "ligne" est l'avant-dernier outil figurant sur la barre d'outils. Lorsque vous cliquez sur l'outil ligne, le curseur devient une croix. Le centre de cette croix est le premier point de votre ligne. Il faut alors définir le deuxième point de la ligne. Cela se fait en laissant le doigt enfoncé sur le bouton gauche de la souris et en tirant le curseur de la souris vers le point de terminaison de la ligne.

La ligne n'est pas sélectionnable : il nous est apparu fort difficile de cliquer sur une ligne. Il n'y a donc pas d'actions définissables sur une ligne. La ligne fera donc office de séparateur (par exemple pour séparer deux parties de l'écran).

Les attributs de la ligne sont la couleur, l'épaisseur et la continuité du trait. Ces caractéristiques sont changeables par double click sur la ligne lors de la création de l'exercice ou lors de sa modification (donc en mode "Moniteur").

5.7.6. *Le circuit...*

Le circuit est la grande nouveauté de cette deuxième version d' "Auteur !". Un circuit se définit grâce à trois zones :

- La zone de circuit : c'est la partie centrale du circuit. Elle est définie avec une taille par défaut. C'est dans cette zone que l'utilisateur est censé maintenir le curseur. Le créateur de l'exercice peut associer des actions à cette zone. Par exemple, tant que le curseur est dans la zone de circuit, on joue un son qui dit : "Tout va bien".
- La zone de dégagement : comme sur les circuits de Formule 1, autour de la piste, il doit y avoir une certaine zone. Cette zone est appelée zone de dégagement. Lorsque le curseur se trouve dans cette zone, le créateur de l'exercice a pu associer une action à la zone et donc, l'utilisateur est prévenu qu'il quitte le circuit.
- La zone de rejet : c'est l'équivalent du rail de sécurité. Lorsque le curseur pénètre dans cette zone, il est censé être rejeté vers la piste. Ici encore, il est possible d'associer des actions.

Toutes ces zones ont, au départ, une largeur par défaut. Un simple double-clic sur le circuit permet un réglage de chaque zone séparément .

5.8. Les actions...

Les actions, comme nous allons le décrire ci-après, sont au nombre de cinq. Elles permettent de jouer des sons, d'animer les animations que vous avez préalablement placées, de changer d'écran,...

Nous allons vous les présenter ci dessous en détail.

5.8.1. Appeler un écran...

L'action "Appeler un écran" est une action très importante, c'est elle qui permet le changement d'écran. Le passage d'un écran à l'autre se fait obligatoirement par l'intermédiaire de cette action. Il n'y a pas d'autres moyens de changer d'écran.

Le changement se fait en signalant le nom de l'écran vers lequel on veut aller. On peut aller vers un écran bien spécifique, ou on peut retourner vers l'écran précédent dans l'enchaînement. Dans ce dernier cas, il faut bien s'assurer que l'écran précédent est bien celui vers lequel on veut aller, ce qui est loin d'être toujours le cas.

Vous n'avez rien à retenir : les noms de tous les écrans sont présentés dans une liste. Cette liste est une liste déroulante. Si l'écran qui figure dans le champ n'est pas celui que vous souhaitez, il vous suffit de cliquer sur la petite flèche juste à côté. La liste des différents écrans apparaît alors, et vous n'avez plus qu'à faire votre choix.

5.8.2. Jouer un son...

L'action "Jouer un son" vous permet de jouer n'importe quel son au format "*.WAV", c'est-à-dire n'importe quel fichier son au format Windows. Pour jouer un son, vous devez le choisir. Un son par défaut existe : c'est le fichier "BOING.WAV". Il suffit de cliquer sur le bouton "Change..." pour en choisir un autre parmi tous les fichiers de ce format que vous possédez. Le nombre de fois que le son sera joué lors de l'exécution de l'action se définit également dans cette boîte de dialogue. Il vous suffit de taper un autre chiffre que celui qui figure dans le champ d'édition présent. Le nombre de fois qu'un son est joué est 1, par défaut.

5.8.3. Attendre...

"Attendre" est une action intéressante. Elle permet de bloquer un exercice, soit pendant un temps déterminé, soit en attendant une action de l'utilisateur. Les actions que l'on peut attendre sont soit un click sur le bouton gauche de la souris, soit la frappe d'une touche du clavier.

Le délai d'attente doit être spécifié en secondes. Attention toutefois, à cause de la gestion du programme par Windows, ce temps risque toujours d'être soit un peu plus court, soit un peu plus long que prévu. Il sera plus long si Windows doit faire d'autres choses plus prioritaires en même temps.

5.8.4. Positionner un objet...

L'action "Positionner un objet" est une action un peu spéciale. Elle permet de déplacer une icône (pour le moment ça ne marche qu'avec des icônes) n'importe où sur l'écran. Pour mettre en oeuvre cette action, vous devez d'abord la sélectionner ; puis il faut l'éditer. Une boîte de dialogue apparaît alors. Dans cette boîte, il y a un champ d'édition (qui est en fait une boîte de liste déroulante, ce qui signifie que vous ne pouvez rien y écrire) : un bouton portant le nom "Positionner" et deux autres champs d'édition qui donnent les coordonnées de l'objet après positionnement. Ces deux derniers champs sont protégés : inutile d'essayer d'y écrire ! La seule façon de les modifier, c'est de cliquer sur le bouton "Positionner".

Quand vous cliquez sur le bouton "Positionner", une autre boîte de dialogue apparaît : il s'agit d'un écran vide, à l'exception de deux boutons. Il vous suffit de cliquer à l'endroit où vous voulez faire apparaître votre icône après le déplacement. La position où vous avez cliqué s'inscrit à l'écran. Cliquez sur le bouton "Ok". La position future de l'objet est maintenant inscrite dans les deux champs de la boîte de dialogue. Il vous suffit, pour valider, de choisir le bouton "Ok". Si vous avez envie de tout annuler, il vous suffit, comme d'habitude, de cliquer sur le bouton "Annuler".

5.8.5. Exécuter une animation...

"Exécuter une animation" est l'action des renforcements positifs ou négatifs par excellence. Mais, avant de pouvoir utiliser cette action, il faut que vous ayez défini au moins une animation complètement. Sans cela, vous ne pourrez pas dire quelle animation il faut exécuter.

La boîte de dialogue qui vous est proposée est très simple. Elle est constituée d'une boîte de liste déroulante et d'un champ d'édition.

La boîte de liste déroulante reprend toutes les animations présentes dans l'exercice. Le champ d'édition contient le nombre de fois que l'on peut exécuter l'animation.

Faites votre choix dans la liste en cliquant sur la petite flèche à côté de la liste. Recliquez sur cette même flèche lorsque votre choix est fait. Il ne vous reste plus qu'à signaler le nombre de fois qu'il faut exécuter l'animation. Il vous suffit, pour cela, de remplacer le chiffre par défaut déjà présent dans le champs d'édition.

Cliquez sur le bouton "Ok" pour valider ou sur "Cancel" pour tout annuler.

5.9. Personnalisation des exercices...

Les exercices que vous créez à l'aide de "Auteur!" sont très généraux. Ils doivent donc être personnalisés en fonction de la personne qui va les faire. Il y a plusieurs façons de personnaliser un exercice. C'est ce que nous allons vous expliquer dans cette section du mode d'emploi.

5.9.1. La personnalisation par les images

Il se peut que les images que vous avez prévues pour un exercice ne soient pas adaptées à toutes les personnes à qui l'exercice est destiné. Il va de soi que vous devez, avant de placer des images, vous renseigner sur les connaissances des personnes exécutantes.

Si vous mettez des pictogrammes d'hommes et de femmes et que vous demandez de chercher l'intrus à quelqu'un pour qui la distinction entre les sexes n'est qu'un concept vide de sens, vous allez être très déçu des résultats de l'exercice car au mieux, la personne ne fera que deviner quel est l'intrus. Elle ne cherchera pas réellement où il est. Tandis que si vous savez que la personne sait bien faire la distinction entre les fruits et les légumes, vous pouvez adapter votre exercice en mettant des fruits et des légumes à la place des hommes et des femmes.

Cela implique que vous ayez sauvé la structure de l'exercice auparavant. Remarquez cependant que si vous ne l'avez pas fait, ce n'est pas grave. Le fait de changer les images puis de le sauver sous un autre nom revient au même que de sauver la structure de l'exercice et puis de changer les images et donnant un autre nom à l'exercice.

5.9.2. La personnalisation par les moyens d'interaction

Autres moyens de personnalisation des exercices, les moyens d'interaction. En effet, il se peut que la souris ne soit pas le moyen le plus approprié pour tout le monde. Il convient donc de savoir avec quel moyen d'interaction le futur utilisateur est familier.

Ici encore, les possibilités sont grandes : vous avez quatre moyens d'interaction à votre disposition. Le premier est le clavier. C'est le moyen par défaut. Mais attention, ne vous méprenez pas, il n'y a pas moyen dans cette version du programme de manipuler les exercices avec le clavier. Dans un avenir proche, nous pourrions manipuler le programme à l'aide des flèches de direction. Celles-ci simuleront les mouvements de la souris. Le clavier est réglable par l'intermédiaire d'une boîte de dialogue permettant de régler la vitesse de répétition d'une touche.

Le second moyen est la souris. C'est LE moyen d'interaction par excellence dans l'environnement Windows. C'est généralement le seul prévu. C'est notre deuxième et dernier moyen d'interaction par défaut.

La souris se règle, elle aussi, par une boîte de dialogue. Nous avons déjà eu l'occasion de décrire comment cela se faisait (Cf. Supra 4.2.6. Le menu INTERACTION).

Deux autres moyens ont été prévus. Quoique peu utilisés en programmation Windows, le joystick et le contact unique sont deux moyens d'interaction nécessaire dans une application telle qu' "Auteur!". Le joystick permet la navigation d'objet et le contact unique permet de simuler le click sur le bouton gauche sans prendre en compte les déplacements de la souris.

Nous avons déjà consacré une partie de ce manuel d'utilisation à l'étude du réglage des moyens d'interaction, nous n'y reviendrons donc pas.

5.9.3. La personnalisation par essais et erreurs

La personnalisation par essais et erreurs est la méthode la plus fréquemment utilisée. Elle consiste à définir un exercice complètement, puis à le faire exécuter par un utilisateur cible. En fonction des retours de celui-ci (par l'intermédiaire du fichier de trace ou par l'observation), le créateur de l'exercice peut le moduler. Il peut aussi le moduler en cours d'utilisation, en employant les différents modes du programme. On peut interrompre un exercice en cours en poussant simultanément sur les touches ALT et M (passage en mode moniteur), faire les modifications qui s'imposent et relancer l'exécution en poussant sur les touches ALT et U (retour au mode "Utilisateur").

Les deux types de personnalisation que nous avons vu ci-dessus peuvent être mis en oeuvre dans le cas d'une personnalisation par essais et erreurs.

Attention, la section suivante ne s'adresse pas aux personnes qui ont lu le sixième chapitre du mémoire intitulé "Mise en place de prérequis par le biais de jeux informatisés". La section suivante est destinée aux utilisateurs qui n'ont pas eu l'occasion de lire le mémoire, afin qu'ils trouvent une méthode de conception d'exercices sans devoir ingérer la centaine de pages du travail.

6.Méthode de construction d'exercices...

6.1. Introduction

Lors de la conception, il nous a été demandé de pouvoir donner une méthode de construction d'exercices pour le logiciel "Auteur!". Cette méthode devait aider les éducateurs et thérapeutes encore effrayés par le diable informatique et son suppôt, l'ordinateur. Mais les aider pour quoi ? Il est vrai que pour construire des exercices avec le programme "Auteur!", il faut souvent raisonner comme un informaticien quand il construit un algorithme. Il y a des choix à faire, des boucles à implémenter,... Bref, toutes sortes de notions purement informatiques.

Nous sommes conscients que ces concepts ne peuvent être abordés sans guidance. Cette guidance, nous nous proposons de la faire par l'intermédiaire de cette "méthodologie". Nous allons montrer comment faire des exercices simples, comment combiner des exercices simples pour faire des exercices plus complexes,... Notre démarche sera la suivante : nous partirons de la conception papier d'un exercice mono-écran, puis nous compliquerons avec l'adjonction d'autres écrans et des enchaînements s'y rapportant. Nous passerons les différentes étapes sur l'ordinateur en expliquant les gestes à faire pour arriver à un résultat plus ou moins satisfaisant.

6.2. A vos crayons...

6.2.1. Un effort d'imagination...

La première étape est un effort d'imagination. En effet, ce n'est pas parce qu'on a un ordinateur que celui-ci va faire tout à votre place. Vous pouvez lancer le programme pendant 237 jours d'affilée, il n'en sortira rien de plus que ce que vous y aurez mis. Il faut donc, premièrement, concevoir les exercices dans votre tête et puis seulement vous asseoir devant... une feuille de papier... Il faut d'abord imaginer l'exercice comme si l'on ne disposait pas d'un ordinateur. Ensuite, on voit si, vu que l'on a un ordinateur, ce genre d'exercice est possible avec "Auteur!". Il faut faire un plan.

6.2.2. Un plan sur papier...

Le plan sur papier est la deuxième étape. Il va vous falloir bien décrire sur une feuille toutes les étapes, tout ce qui sera des écrans, des icônes, des actions,... Il faut également écrire les enchaînements entre les éléments. Nous vous proposons de suivre la démarche suivante : lorsque l'exercice est clair dans votre tête, faites-en une maquette sur papier. Vous verrez alors les petits défauts auxquels vous n'aviez pas pensé.

Pour reprendre un exemple bien connu, nous allons développer, avec vous, un exercice : c'est l'exercice de l'intrus. C'est un exercice simple, que tout le monde connaît. L'effort d'imagination n'est donc pas trop grand.

Maintenant que l'exercice est clair dans la tête de chacun, nous allons poser ses bases sur papier. Il nous faut donc répondre à certaines questions : quels pictogrammes va-t-on employer ? Qu'est-ce que l'enfant connaît ? Connaît-il les couleurs ? les formes ? Où va-t-on placer les pictogrammes sur l'écran ? en haut, au milieu, en bas, ... ? De quelle couleur doit être la couleur du fond de l'écran ? ...

Il faut donc régler tous les petits détails. Dans un premier temps, nous n'utiliserons qu'un seul écran. Une nouvelle question apparaît : si on n'utilise qu'un seul écran, comment va-t-on faire les renforcements positifs et négatifs ? Voyons cela en détail.

Nous voulons faire un exercice mono-écran reprenant le principe du jeu de l'intrus. Nous allons placer les dessins au milieu de l'écran en une ligne. Le fond de l'écran devrait avoir une couleur neutre (blanc ou blanc cassé). Nous allons établir un jeu de l'intrus où l'intrus sera une personne de sexe différent. Il y aura trois dessins de femme et un dessin d'homme. nous supposons que l'enfant ne connaît pas les couleurs : nos dessins seront tous bleutés. Nous supposons que l'enfant est capable de différencier un homme d'une femme, ne serait-ce que par l'aspect vestimentaire.

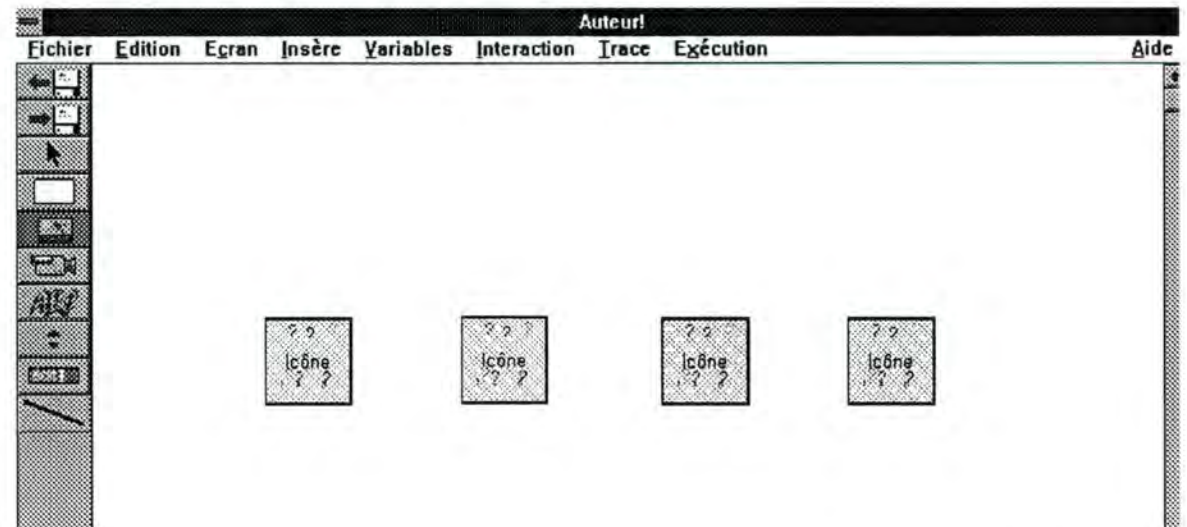
Pour les renforcements, nous prévoyons un son associé à chaque pictogramme. Ces sons seront respectivement : un cri de joie pour la réussite, une expression de déception pour l' échec. Nous pouvons ainsi nous limiter à un seul écran.

Maintenant que tout est défini, asseyons-nous devant la machine et construisons cet exercice. Cliquer deux fois sur l'icône de "Auteur!" est la première étape. Vous êtes directement dans le mode de conception.

6.3. Passons sur la machine (un exercice mono-écran)....

Nous voilà devant l'écran de conception d' "Auteur!". C'est d'ici que nous allons, ensemble, construire l'exercice de l'intrus que nous avons conçu dans notre tête et sur papier. Nous allons procéder en 3 étapes.

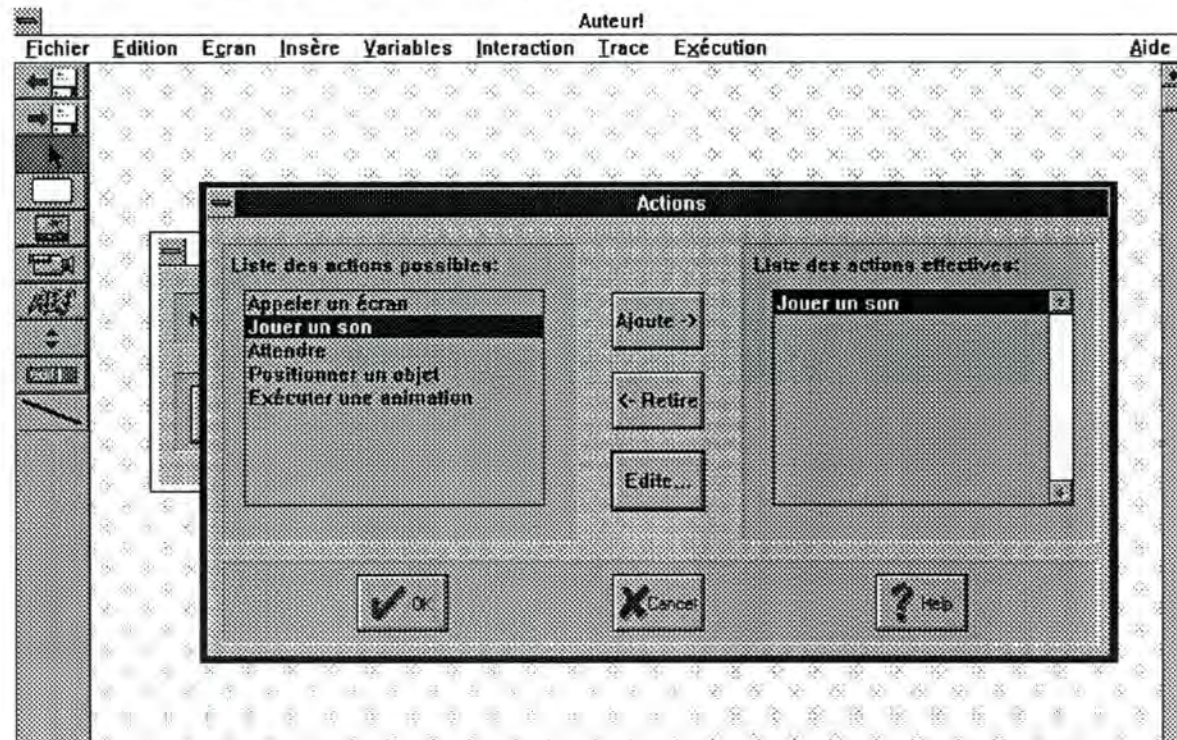
La première de ces étapes est de placer 4 icônes. Pour cela, sélectionnez l'outil "icône". C'est celui avec un petit dessin représentant une île perdue sur un océan et un petit soleil en haut à gauche. Le curseur de la souris change alors et prend la forme de l'outil (Ne vous effrayez pas, c'est normal). Vous allez placer 4 icônes en ligne, comme prévu dans la version papier de l'exercice, vous devez obtenir quelque chose dans le genre de ce qui suit :



Les quatre carrés jaunes sont les quatre futurs pictogrammes. Pour moment, ils n'affichent pas encore les images que nous avons décidés d'y mettre. C'est normal, car lorsque l'on crée une icône, celle-ci est affichée avec un pictogramme par défaut. Nous changerons ces pictogrammes lors d'une étape ultérieure.

La seconde étape va nous faire modifier le fond d'écran. Nous allons en profiter pour adjoindre une action à l'écran. Il devra jouer un son qui reprend le but de l'exercice. Pour faire ces modifications, sélectionnez la flèche dans la barre d'outils. Il faut cliquer deux fois sur le fond. Choisissez le bouton "Couleur" et donnez à l'écran la couleur que vous souhaitez pour le fond de l'exercice (n'oubliez pas que nous avons défini une couleur neutre sur papier). Une fois que cela est fait, choisissez le bouton "Action". Dans la boîte de dialogue qui s'ouvre, sélectionnez l'action "Jouer un son" puis appuyer sur le bouton "Ajoute ->" et ensuite sur le bouton "Edite". Choisissez le fichier qui correspond à l'explication du jeu, puis cliquez sur "Ok".

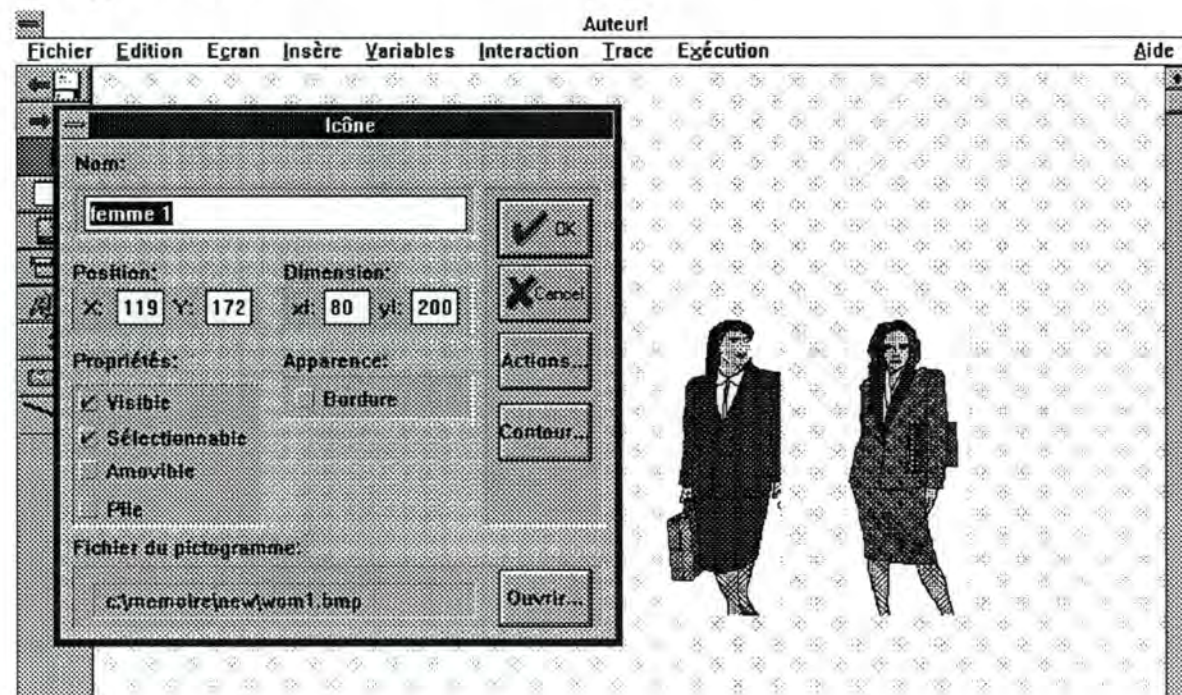
Vous devez revenir à la situation suivante :



Remarquez que le fond d'écran est déjà dans la couleur que vous avez choisie. Maintenant que nous avons changé les paramètres de l'écran, nous allons pouvoir changer les paramètres des icônes. Au cours de cette troisième étape, nous allons pouvoir changer le dessin et associer une action à chaque icône. Pour des raisons évidentes de répétitions inutiles, nous ne ferons la manoeuvre avec vous qu'une seule fois.

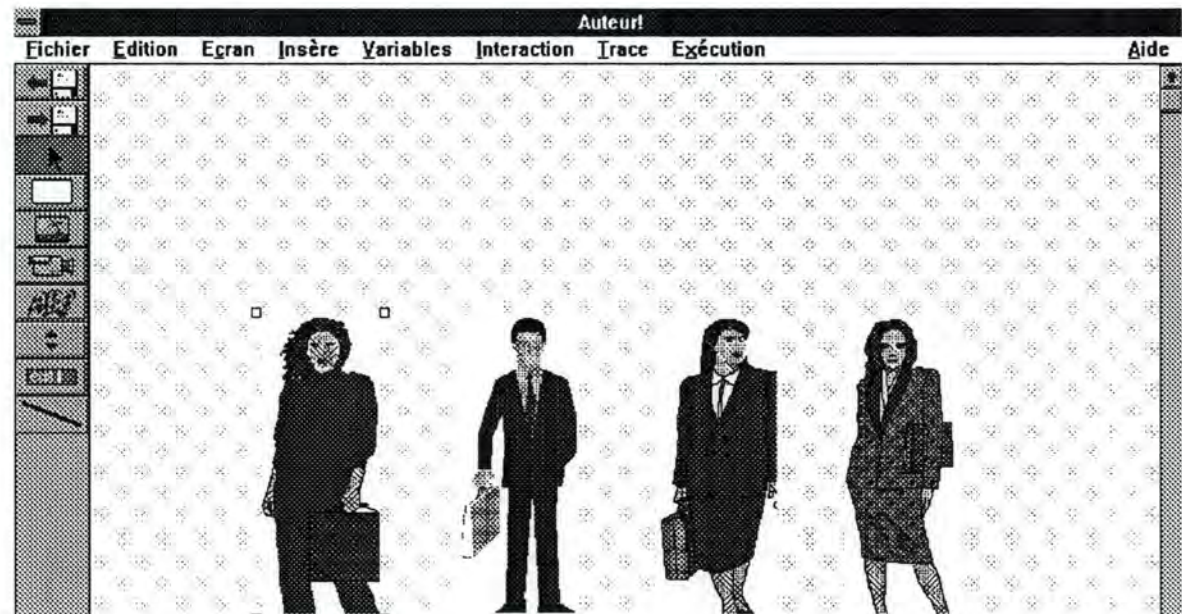
Notez aussi que l'on peut changer le nom de l'écran et lui donner un nom plus significatif ; Que pensez-vous de "Intrus 1" ? Au moins, comme ça, on sait de quoi on parle.

La troisième étape débute par un double click sur une des icônes. La boîte de dialogue suivante apparaît alors :



Nous vous proposons de renommer les icônes : vous aurez ainsi les icônes "Femme 1", "Femme 2", "Femme 3", "Homme 1". Nous en sommes à l'icône "Femme 1". Une fois que vous l'aurez renommée, cliquez sur le bouton "Actions...". Une nouvelle boîte de dialogue apparaît. Comme pour l'écran, choisissez l'action "Jouer un son". Editez cette action après l'avoir ajoutée dans la liste. N'oubliez pas que l'intrus n'est pas cette icône. Donc si l'enfant clique sur "Femme 1", il devrait entendre un son exprimant la déception. Choisissez donc bien le son que vous allez mettre. Quand vous aurez choisi, appuyez deux fois sur les boutons "Ok". Vous devez vous retrouver dans la boîte de dialogue d'édition d'une icône. Une fois là, appuyez sur le bouton "Ouvrir..." et choisissez le pictogramme dont le nom est "WOM1.BMP". Appuyez sur "Ok", et encore une fois sur "Ok".

Si vous faites cela pour toutes les icônes respectivement avec les fichiers "WOM2.BMP" pour l'icône "Femme2", "WOM3.BMP" pour l'icône "Femme 3" et "MAN1.BMP" pour l'icône "Homme 1", vous devriez obtenir, à peu de choses près, l'écran suivant :



Il ne vous reste plus qu'à tester l'exercice. Pour cela, choisissez l'option "Exécution" du menu. Vous verrez alors le menu et la barre d'outils disparaître. L'exercice se fera en plein écran. Vous pouvez essayer votre création : un click sur une icône représentant une femme doit amener un son exprimant la déception, le click sur l'icône représentant un homme doit amener un son exprimant la réussite.

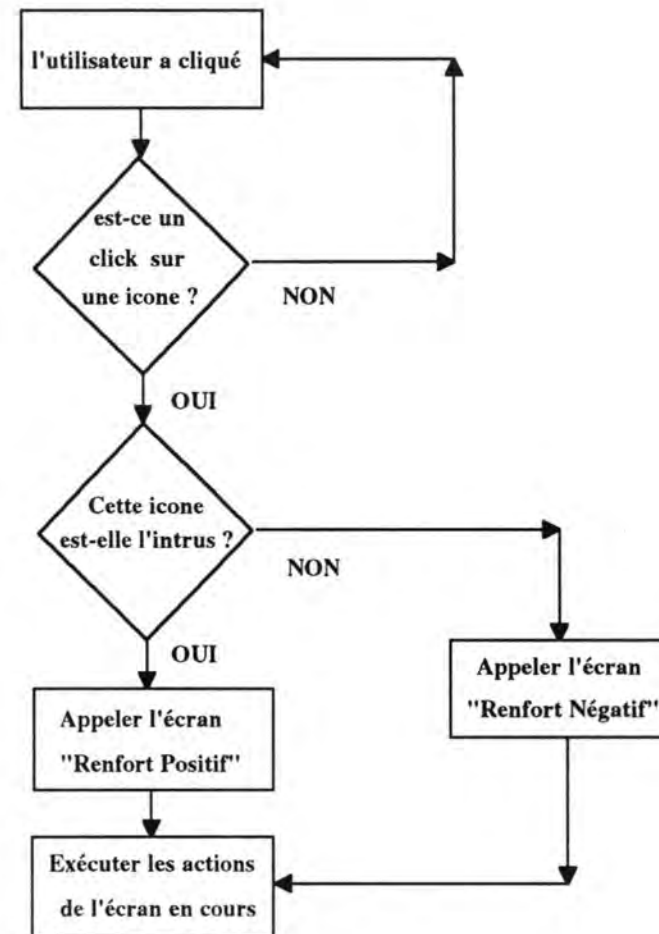
6.4. A vos crayons (un exercice multi-écrans)...

Maintenant que nous maîtrisons bien la technique de création d'un exercice sur un seul écran, voyons comment nous allons pouvoir ajouter des écrans supplémentaires. Ces écrans, sur papier, serviront de renforcements positifs et négatifs. Nous n'allons donc ajouter que deux écrans. L'un sera de couleur verte et reprendra le son exprimant la réussite. Il comportera un pictogramme montrant à l'enfant qu'il a réussi l'exercice. Pour éviter à l'enfant des manipulations inutiles, le retour à l'écran principal de l'exercice se fera automatiquement après 5 - 10 secondes. L'autre écran sera de couleur rouge, il affichera aussi un dessin montrant à l'enfant que sa tentative a échoué. Il reprendra également le son exprimant l'échec, la déception. De même que pour l'écran de renforcement positif, le retour à l'écran principal du jeu se fera automatiquement.

Nous appellerons le premier de ces écrans "Renfort Positif" et le second "Renfort Négatif". Ainsi, nous ne pourrions les confondre. L'écran "Renfort Négatif" servira dans les trois cas où l'enfant clique sur la mauvaise icône.

Nous pensons qu'ainsi, l'exercice est parfaitement défini. De plus il est très général car il suffit de changer les pictogrammes dans l'écran principal pour avoir un nouvel exercice.

Sur papier, l'exercice peut se représenter comme suit :



Voyons ce que cela donne sur machine.

6.5. Passons sur machine (un exercice multi-écrans)...

Ici aussi, nous allons travailler par étape. Nous repartons pour cela du résultat de la dernière étape de la conception d'un exercice mono-écran. La première chose à faire, est d'éditer chaque icône, de manière à enlever l'action "Jouer un son". Nous allons remplacer ces actions par d'autres, mais cela doit se faire plus tard.

Ensuite, quand plus aucune icône ne possède d'action, il vous faut définir les nouveaux écrans. Pour créer un nouvel écran, choisissez le petit rectangle blanc dans la barre d'outils. Vous remarquerez que lorsque vous cliquez sur ce bouton, il ne se passe, en apparence, rien. En fait, le fait d'appuyer sur ce bouton, crée un nouvel écran *derrière* l'écran en cours.

Pour créer les deux écrans de renforcement, vous allez devoir cliquer deux fois sur cet outil. Pour vérifier que la création s'est bien faite, allez dans le menu et choisissez l'élément "Ecran". Si tout s'est bien passé (et il n'y a aucune raison que cela ne se passe pas bien), vous devez avoir trois noms d'écrans dans ce menu déroulant :

- ♦ "Intrus 1" qui est le nom de votre premier écran.
- ♦ "Ecran 2" qui résulte du premier click sur l'outil écran.
- ♦ "Ecran 3" qui résulte du second click sur l'outil écran.

Avant de passer sur un autre écran, nous allons compléter l'écran "Intrus 1" en assignant des actions aux icônes. Double cliquez sur chacune des icônes à son tour, et appuyez sur le bouton "Action" pour chacune d'elles. La colonne de droite de la boîte de dialogue "Action" est normalement vide puisque vous avez effacé les actions "Jouer un son". Vous allez donc choisir, dans la colonne de gauche, l'action "Appeler un écran". Ce sera la seule action que nous assignerons à une icône.

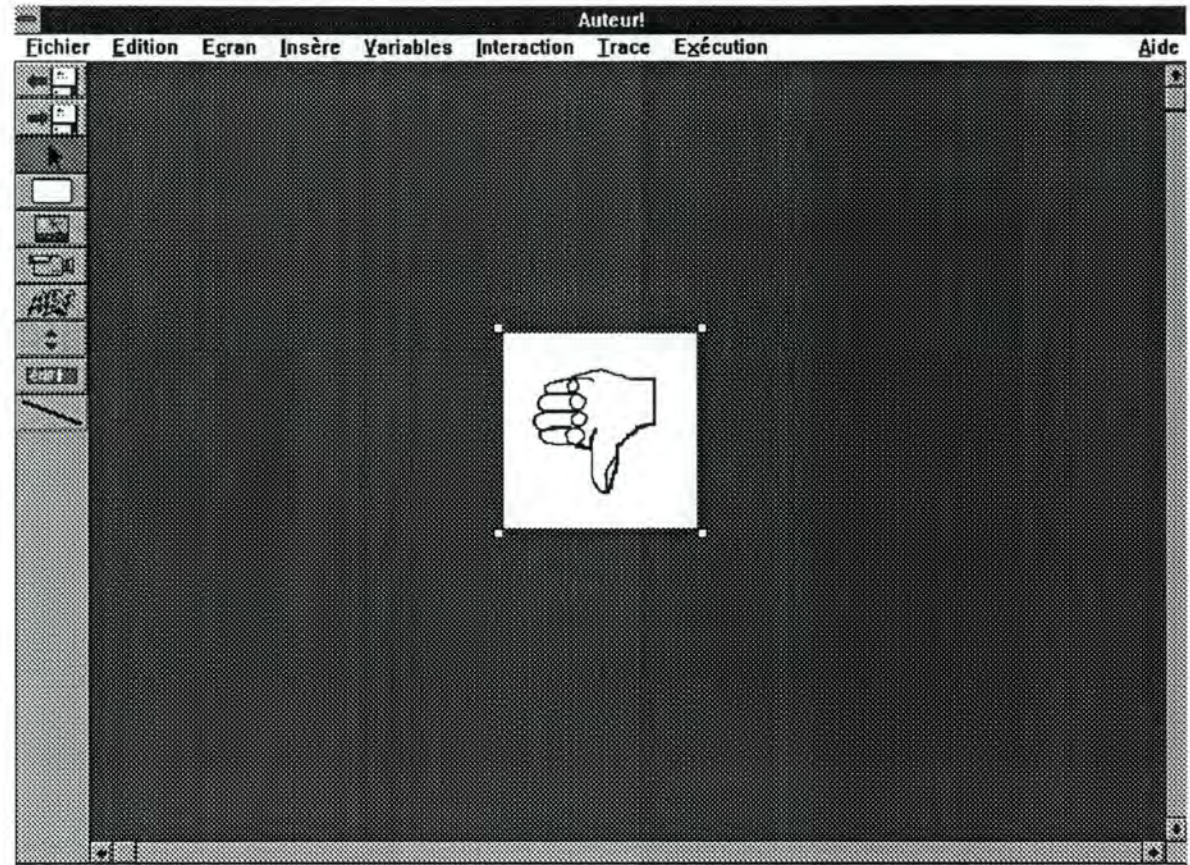
Nous allons maintenant passer à la modification des deux autres écrans. Dans le menu "Ecran", cliquer sur l'élément "Ecran 2". Un écran vide doit apparaître. Sur cet écran vide, placer une icône au milieu. Double cliquez sur l'icône afin d'en changer le pictogramme. N'oubliez pas que nous avons décidé de placer des pictogrammes exprimant soit la réussite, soit l'échec de l'exercice. Choisissez donc judicieusement.

Une fois cette modification faite, appuyer sur "Ok". Votre icône est placée. Supposons que l'écran "Ecran2" sera l'écran de renforcement négatif. Il vous faut maintenant changer la couleur de fond de manière à l'amener au rouge. Il vous faut aussi définir des actions pour l'écran.

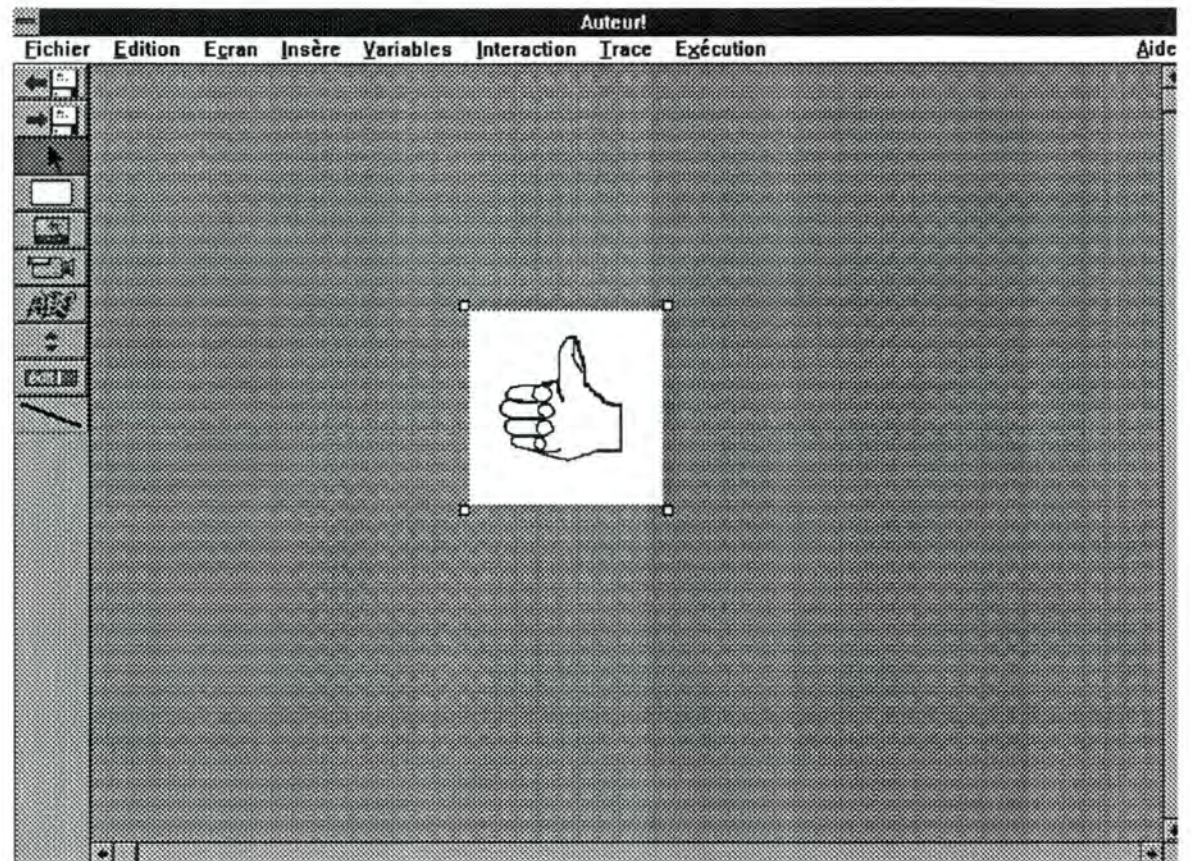
Voici ce que nous vous proposons de faire : Sélectionnez la flèche dans la barre d'outils. Double cliquez sur le fond de l'écran, choisissez le bouton "Couleur". Choisissez la couleur rouge. Appuyez sur "Ok". Sélectionnez le bouton "Action". Dans la colonne de gauche, choisissez "Jouer un son", ajoutez-la à la liste. Choisissez l'action "Attendre" et ajoutez-la à la liste. Editez le son de manière à pouvoir reprendre le son que nous avons utilisé précédemment. Editez l'action "Attendre". Enlevez les marques éventuelles dans les cases à cocher (sorte de petits losanges), et spécifiez un délai d'attente (dans ce cas, 7 secondes). Les deux actions seront effectuées l'une après l'autre.

N'oubliez pas de renommer votre écran : son nom devra être "Renfort Négatif".

Vous pouvez procéder de la même façon pour l'autre écran, celui de renforcement positif. Donc en plus de l'écran principal de l'exercice, vous devez avoir deux écrans ressemblant à ceux-ci :



Ceci étant l'écran "Renfort Négatif". L'écran "Renfort Positif" est le suivant : (voir page suivante).



Voilà votre exercice fin prêt à être exécuté. Attention toutefois : il faut bien veiller à ce que l'enfant comprenne l'exercice que vous voulez lui faire faire.

6.6. La méthode par essais et erreurs...

C'est ici que la méthode par essais et erreurs joue son rôle. Imaginons que vous ayez proposé cet exercice tel quel à un enfant, pour qui les concepts de différence entre les sexes ne sont pas établis. Il ne verra pas la différence entre un homme et une femme alors qu'il la verrait entre un homme et un poulet.

Le programme "Auteur!" vous a permis de définir un exercice très général. En effet, il vous suffit de changer les icônes que vous avez placées dans l'exercice. Plutôt que de mettre des femmes et un homme, vous essaieriez des fruits et un animal, des jouets connus par l'enfant et une tondeuse à gazon,...

De plus, le moyen d'interaction peut très bien ne pas convenir. Encore une fois pas de panique, il vous suffit d'interrompre l'exercice en cours (par les touches ALT + M) et de reparamétrer l'exercice en fonction des aptitudes de l'enfant. Le paramétrage du moyen d'interaction se fait par l'intermédiaire de l'élément "Interaction" du menu.

Vous pouvez aussi évaluer votre exercice à posteriori, en regardant la trace laissée par celui-ci. Il vous suffit d'éditer le fichier "TRACE.TRA". Mais pour cela, vous devrez dire au programme qu'il doit enregistrer les actions de l'utilisateur. Cela se fait par l'intermédiaire de l'élément "Trace" du menu.

6.7. Conclusion

En conclusion de cette partie, nous dirions que construire un exercice avec "Auteur!" n'est pas difficile. Il suffit de se familiariser avec certains concepts, certains modes de raisonnement.

Bien sûr, nous n'avons, par cet exercice très connu qu'est le jeu de l'intrus, montré qu'une partie élémentaire de ce qu' "Auteur!" peut faire. Mais les concepts de base sont ainsi posés. L'imagination des éducateurs et des thérapeutes fera le reste.

7. Appendice A : Ajouter des sons dans la bibliothèque...

Ajouter des sons dans la bibliothèque de sons qui vous est déjà offerte n'est pas difficile. N'importe quelle carte sonore est capable d'enregistrer des sons de plus ou moins bonne qualité. Sous Windows, vous avez le choix entre deux formats : le format VOC et le format WAVEFORM (WAV). Le programme "Auteur !" est conçu pour fonctionner avec des fichiers sonores au format WAVEFORM (WAV). N'importe quel fichier sonore enregistré dans ce format peut donc convenir.

Pour les utilisateurs qui veulent récupérer des sons déjà préenregistrés mais dans un autre format, il existe des "convertisseurs de format". Ce sont des petits programmes qui prennent un fichier sonore, par exemple au format VOC et qui le transforment en un fichier sonore au format requis (WAV).

Une fois que vous avez ces fichiers sonores, il vous suffit de les ajouter dans le répertoire des fichiers sonores d' "Auteur !".

8. Appendice B : Ajouter des images dans la bibliothèque...

Comme pour les sons, c'est encore une fois une question de format. Le format accepté par le programme "Auteur !" est une format commun à bien des applications Windows. Il s'agit du format BITMAP (BMP), défini en standard sous Windows. N'importe quel fichier au format BMP est donc accepté par "Auteur !".

Si vous avez des fichiers graphiques qui ne sont pas dans ce format, un programme comme "Graphic Workshop" devrait pouvoir vous aider puisqu'il permet des conversions d'images dans différents formats (et notamment au format BMP).

Une fois les images converties, il suffit de les placer dans le répertoire des images prévues pour "Auteur !".

9. Appendice C : Troubles du fonctionnement...

Cette section est consacrée aux troubles de fonctionnement dont nous sommes avertis. Il va sans dire que cette liste (très courte) n'est absolument pas exhaustive. Il vous appartient, à vous, utilisateurs, de nous communiquer les éventuelles malfunctions du logiciel, et ce afin que nous puissions y remédier au plus tôt.

Le programme s'arrête quand on veut ajouter une icône ou une animation sur l'écran.

Ce problème est sans doute dû au fait que les fichiers d'images par défaut, IM.BMP et ANIM.BMP, ne sont pas placés au bon endroit sur le disque dur. Cette version d' "Auteur !" doit malheureusement se situer dans un répertoire "C:\AUTEUR". Et de même, les deux fichiers précités doivent s'y trouver aussi. Nous espérons pouvoir corriger ce fait dans une version ultérieure.

Pour remédier au problème, il faut sortir d' "Auteur !", et vérifier qu'il existe bel et bien un répertoire C:\AUTEUR qui comprend non seulement le fichier exécutable du programme, mais aussi les deux fichiers IM.BMP et ANIM.BMP. Si ces deux fichiers ne s'y trouvent pas, il faut chercher où ils se trouvent. Si vous les avez effacés par erreur, il faut reprendre les disquettes d' "Auteur !" et les recopier dans le bon répertoire.

10. Index...

A

Actions, 27

Appeler un écran, 27

Attendre, 28

Exécuter une animation, 29

Jouer un son, 28

Positionner un objet, 29

Ajouter des images dans la bibliothèque..., 48

Ajouter des sons dans la bibliothèque..., 48

Animation, 23

Auteur

Programmes, 2

Auteur !, 3

Du programme, 17

B

Barre de statut, 18

Boîtes de dialogue, 19

C

Carte sonore, 1

Circuit, 26

Clavier, 12

Configuration

requis, 1

matérielle, 1

logicielle, 1

D

Démarrer, 4

Driver, 2

E

Ecran

Capacité, 1

Outil, 24

I

Icône, 21

Installation, 4

J

Joystick, 14

L

Ligne, 25

M

Menus

- Aide, 17
 - Ecran, 10
 - Edition, 9
 - Effacer, 9
 - Sélectionner tout, 9
 - Execution, 16
 - Fichier, 5
 - Enregistrer, 6
 - Enregistrer écran, 7
 - Nouveau, 5
 - Ouvrir, 5
 - Ouvrir écran, 6
 - Préférences, 8
 - Quitter, 8
 - Insère, 10
 - Interaction, 11
 - Trace, 16
- Méthodes de construction d'exercices, 34
- Mode moniteur, 20
- Mode utilisateur, 20

O

- Outils, 20

P

- Personnalisation des exercices, 30
 - Par essais et erreurs, 32
 - Par les images, 30
 - Par les moyens d'interaction, 31

R

- Ruban d'outils, 18

S

- Souris, 13
- Switch, 14

T

- Texte, 25
- Troubles du fonctionnement, 49